OPTIMAL CUT-POINTS FOR DIAGNOSTIC VARIABLES IN COMPLEX SURVEYS

by

SAMAR ADNAN MADI

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN APPLIED MATHEMATICAL SCIENCES

2023

Oakland University
Rochester, Michigan

Doctoral Advisory Committee:

Dorin Drignei, Ph.D., Chair
Elise Brown, Ph.D.
Theophilus Ogunyemi, Ph.D.
Subbaiah Perla, Ph.D.
Hon Yiu (Henry) So, Ph.D.

ABSTRACT

OPTIMAL CUT-POINTS FOR DIAGNOSTIC VARIABLES IN COMPLEX SURVEYS

by

Samar Adnan Madi

Advisor: Dorin Drignei, Ph.D.

The ability to diagnose an individual is crucial in promoting treatment and improved health. However, finding a simple tool to base the diagnosis on can be complicated. This research will focus on developing statistical methodology for accurate diagnostic tests in the context of complex survey data. The proposed method will be illustrated with data from National Health and Nutrition Examination Survey (NHANES) to construct a diagnostic test to predict cardiometabolic disease risk in the US younger population. This research will begin with the exploration of a single diagnostic variable to be used as a diagnostic tool. The first 1-dimensional method explored uses receiver operating characteristic (ROC) curves for survey data as a means of determining an optimal cut-point for the diagnostic variable. This method is shown to be accurate but not conducive to multi-variable diagnostic tools using survey data. Another 1-dimensional method uses logistic regression for survey data to determine an optimal cut-point, using minimizing information criteria such as AIC to select the cut-point. The method is applied to NHANES data but considering a single diagnostic variable is shown to be too simplistic to create a comprehensive diagnostic tool. This method will then be extended to a multi-dimensional case, creating a diagnostic tool based on multiple variables using

logistic regression for survey data. This method, although accurate, is shown to be time-consuming and computationally inefficient. A modified method using kriging-based optimization is proposed. Under this method, a more efficient search algorithm of efficient global optimization is explored, using a criterion of expected improvement. This proposed method is more computationally efficient in creating a multi-dimensional diagnostic tool. Application of these methods in a healthcare setting could be beneficial in promoting quick and easy diagnosis.

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Motivation

Modern medicine has advanced greatly over time, with new and effective methods in diagnosing issues in patients and promoting treatment options. However, an early diagnosis of a possible health issue is potentially the most important factor in preventative care. To develop this diagnosis, data-based tools can help detect health complications in their early stages. Accurately diagnosing a patient early on is crucial, as it gives the patient and physician the best chance of finding a treatment plan to promote a better health outcome as quick as possible.

In diagnosing an individual, having a concise test to make an accurate decision is of major importance. In a clinical setting, this test is usually built on one or multiple continuous biomarkers, which are used as proxies to determine the diagnosis. For these biomarkers, cut-points are established that can be used as thresholds for the diagnosis of a binary outcome. Then the diagnosis of diseased or non-diseased can be based on whether an individual falls below or above that cut-point.

Currently, there are a few methods used to determine an optimal cut-point that would provide the most accurate diagnostic test. Since a diagnostic test may consist of several threshold values, having a method that can provide both an accurate and a timely diagnostic tool is vital. However, some of these methods have not been well-researched with complex survey designs. This study will both modify existing methods and develop new methods to determine diagnostic tests in the context of survey data.

1.2 Thesis Outline

The focus of this research is to develop statistical methodology for diagnostic tests to classify each subject into one of two categories, e.g. diseased or not, when data are collected from complex surveys. These diagnostic tests will be built around one or multiple diagnostic variables, for which thresholds (or cut-points) will be determined. Several methods will be applied and compared based on their efficiency and accuracy in developing this diagnosis. Specifically, receiver operating characteristic curves and information criteria such as the Akaike information criterion (AIC) will be used, either directly when computationally feasible or through kriging-based optimization to mitigate the computational burden.

In Chapter Two, basic survey methods are reviewed. Additionally, the NHANES dataset and survey design is explained. An in-depth review of kriging methods used to find diagnostic thresholds on a multi-dimensional level is provided. This chapter will provide a setup of the work that will follow later in the research.

In Chapter Three, receiver operating characteristic curves for complex survey data and the associated optimal cut-points are developed to find the diagnostic threshold of a single diagnostic variable (i.e. 1-dimensional case). This chapter will also discuss previous literature on ROC curves. The novel approach developed will be applied to the NHANES dataset.

In Chapter Four, the 1-dimensional case is revisited using a logistic regression direct approach based on information criteria and survey data. The advantages and

disadvantages of this approach are explored and applied to the NHANES dataset. In addition, a simulation study is run to demonstrate the accuracy of this direct method.

In Chapter Five, the direct method of Chapter Four is expanded to multiple diagnostic variables, but quickly shown to be computationally inefficient to build a multi-dimensional diagnostic tool. The proposed method based on kriging-based optimization of information criteria is applied and explored to show its benefits in efficiency and accuracy. The proposed method uses the concept of efficient global optimization along with an expected improvement. To show the feasibility of this process, a simulation study under kriging-based optimization is presented. The method is then applied to the NHANES data.

Finally, Chapter Six concludes the advantages and limitations of each of the proposed methods and covers potential ideas for improvement.

CHAPTER TWO

PRELIMINARIES

2.1 Survey Methods

2.1.1 Survey Data

The research in this thesis is applied to a dataset collected using a complex survey. With simple random sampling (SRS), sampled data is collected from a population of respondents so that each sample of the same size has equal probability of being selected. Survey data, on the other hand, has a more complex structure, since for example respondents are selected from the population by survey methods that involve similarly categorized groups. In survey sampling, different respondents may have different probabilities of selection, and are thus assigned a weight conceptualizing how representative of the population they are. A lower weight indicates that the respondent represents fewer individuals in the population, thus increasing the probability of selection and representativeness in the sample of that group. Clustering and stratification are two basic survey designs and together constitute the building blocks of more complex surveys. More details can be found in Lohr (2010).

In clustering, the population is broken down into groups categorized by a feature that induces similarity among the individuals of the same group. Each group is considered a cluster, and the set of clusters encompass the population. For example, a population set of a city could be broken down into clusters by considering each street as one cluster. Clustering can also be done in multiple stages, with each stage being defined as a sampling unit. The primary sampling unit (PSU) is the largest cluster type. In the city case, each street would be a PSU. Then in each PSU, secondary sampling units (SSU) can

be defined as a second stage of clustering. For example, in each street PSU, the SSUs might be each household. This order of clustering can be broken down into more stages as far as each survey design allots for. A sample is then pulled by selecting a certain number of PSUs from the total population of PSUs. Next, within each selected PSU, a certain number of SSUs are selected. This is repeated per sampling unit. Within each stage of sampling, a number of methods, such as simple random sampling, can be implemented to select the final sample. Thus, clustering allows for groups of data to be blocked off together, and individuals in a select number of groups are surveyed.

In contrast, stratification allows for a sample from each group of a partitioned population to be selected. Like clustering, stratification groups data together into groups called "strata" that share a common feature. In stratified sampling, the population is divided into disjoint subgroups based on certain characteristics of the population (such as race, age, or gender) and then individuals are randomly selected within each of the different strata. While clustering selects from only some PSUs at the beginning stage of sampling, stratification selects a sample from every stratum. This ensures that each stratum of the population is represented in the sample. In the city example, a possible stratification could be made by dividing the city into several strata delimited by major roads. Then, individuals would be selected from each stratum, to ensure that all such city areas were represented in the sample.

Both methods of survey design have their advantages. Cluster sampling is beneficial when, for example, it is more cost-effective to sample from the same cluster than sampling randomly from the population using a simple random sample. This way,

the population can be broken down into groups, and only some of those groups will be sampled. Stratification, on the other hand, would require sampling from every stratum. Generally, the number of strata tends to be much smaller than the number of clusters. Clustering is less costly and more efficient but stratified sampling allows for more precise results as data is pulled from every stratum. Generally, complex surveys such as NHANES includes both types of survey designs, as described further in section 2.2.

2.1.2 Stratification

For stratified random sampling, let N be the total number of units in the population, partitioned into H strata. Each stratum h has N_h sampling units. Then

$$N = N_1 + N_2 + \ldots + N_H \tag{2.1.1}$$

In order to pull a sample, independently take a simple random sample from each stratum. Let n be the total number of units in the sample. Then similarly,

$$n = n_1 + n_2 + \ldots + n_H \tag{2.1.2}$$

Let y_{hj} be the value of the j^{th} unit in stratum h and n_h be the number of units randomly selected from stratum h, with S_h being the set of n_h units. Then then sample mean in stratum h can be estimated by

$$\bar{y}_h = \frac{1}{n_h} \sum_{j \epsilon S_h} y_{hj} \tag{2.1.3}$$

and the sample variance in stratum h by

$$s_h^2 = \sum_{j \epsilon S_h} \frac{(y_{hj} - \bar{y}_h)^2}{n_h - 1} \tag{2.1.4}$$

Then the overall population mean can be estimated by

$$\bar{y}_{str} = \sum_{h=1}^{H} \frac{N_h}{N} \bar{y}_h \tag{2.1.5}$$

which is a weighted average of the sample stratum averages, and its variance can be estimated by

$$\hat{V}(\bar{y}_{str}) = \sum_{h=1}^{H} (1 - \frac{n_h}{N_h})(\frac{N_h}{N})^2 \frac{s_h^2}{n_h} \tag{2.1.6}$$

2.1.3 Clustering

In one-stage clustering, all SSUs are sampled from the selected PSUs. In two-stage clustering sampling, however, only a select number of SSUs are selected within the selected PSUs. To achieve this, let the total population have N PSUs. First, an SRS S of n PSUs is taken. Then, an SRS of SSUs is taken from each selected PSU. Let M_i be the number of SSUs in the i^{th} PSU and m_i be the number of SSUs in the sample from the i^{th} PSU. Then y_{ij} denotes the measurement for the j^{th} element in the i^{th} PSU, and S_i denotes the SRS of m_i elements from the i^{th} PSU.

Let k be the total number of SSUs in the population, so

$$k = \sum_{i=1}^{N} M_i \tag{2.1.7}$$

The estimate for the total in the i^{th} PSU is given by

$$\hat{t}_i = \sum_{j \epsilon S_i} \frac{M_i}{m_i} y_{ij} \tag{2.1.8}$$

and the sample mean per SSU for PSU i is given by

$$\bar{y}_i = \sum_{j \epsilon S_i} \frac{y_{ij}}{m_i} \tag{2.1.9}$$

Then, an unbiased estimator of the population total is given by

$$\hat{t}_{unb} = \frac{N}{n} \sum_{i \epsilon S} M_i \bar{y}_i \tag{2.1.10}$$

with variance

$$\hat{V}(\hat{t}_{unb}) = N^2 \left(1 - \frac{n}{N}\right) \frac{s_t^2}{n} + \frac{N}{n} \sum_{i \epsilon S} (1 - \frac{m_i}{M_i}) M_i^2 \frac{s_i^2}{m_i} \tag{2.1.11}$$

where

$$s_t^2 = \frac{\sum_{i \epsilon S} (\hat{t}_i - \frac{\hat{t}_{unb}}{N})^2}{n - 1} \tag{2.1.12}$$

$$s_i^2 = \frac{\sum_{j \epsilon S_i} (y_{ij} - \bar{y}_i)^2}{m_i - 1} \tag{2.1.13}$$

and an unbiased estimator of the population mean is given by

$$\hat{\bar{y}}_{unb} = \frac{\hat{t}_{unb}}{M_0} \tag{2.1.14}$$

with variance

$$\hat{V}(\hat{\bar{y}}_{unb}) = \frac{\hat{V}(\hat{t}_{unb})}{M_0^2} \tag{2.1.15}$$

where M_0 is the total number of elements in the population.

2.1.4 Ratio Estimation

This research will also incorporate ratio estimation, which is useful in a number of scenarios: estimating the population total when the population size N is unknown,

adjusting estimates from the sample to reflect demographic totals, adjusting for non-response, etc. For ratio estimation, both y_i and x_i (the auxiliary variable) need to be measured in each sample unit. For a population of size N, let the totals for x and y be

$$t_y = \sum_{i=1}^{N} y_i \ , t_x = \sum_{i=1}^{N} x_i \tag{2.1.16}$$

Then their ratio is

$$B = \frac{t_y}{t_x} \tag{2.1.17}$$

When applied to stratified sampling, the ratio estimator is given by

$$\hat{B} = \frac{\hat{t}_{y,str}}{\hat{t}_{x,str}} \tag{2.1.18}$$

where

$$\hat{t}_{y,str} = \sum_{h=1}^{H} N_h \bar{y}_h \ , \hat{t}_{x,str} = \sum_{h=1}^{H} N_h \bar{x}_h \tag{2.1.19}$$

Its variance can be estimated using Taylor linearization or replication methods, which will be discussed in the next chapter. In the combined ratio estimator, the strata are combined first to estimate the totals, and then ratio estimation is applied (Lohr, 2010). An application of ratio estimation is to obtain alternative estimators for the population total $\hat{t}_{yrc} = \hat{B} t_x$, with variance estimated by

$$\hat{V}\left(\hat{t}_{yrc}\right) = \left(\frac{t_{x,str}}{t_{y,str}}\right)^2 \hat{V}\left(\sum_{h=1}^{H} \sum_{j \epsilon S_h} w_{hj} \, e_{hj}\right) \tag{2.1.20}$$

where

$$e_{hj} = y_{hj} - \hat{B} x_{hj}. \tag{2.1.21}$$

When ratio estimation is applied to two-stage cluster sampling in the case where the x's are the PSU sizes, the ratio estimator for the population mean is

$$\hat{\bar{y}}_r = \frac{\sum_{i\epsilon S} \hat{t}_i}{\sum_{i\epsilon S} M_i} = \frac{\sum_{i\epsilon S} M_i \bar{y}_i}{\sum_{i\epsilon S} M_i}, \tag{2.1.22}$$

Its variance is estimated by

$$\hat{V}(\hat{\bar{y}}_r) = \frac{1}{\bar{M}^2}\left[\left(1 - \frac{n}{N}\right)\frac{s_r^2}{n} + \frac{1}{nN}\sum_{i\epsilon S} M_i^2 \left(1 - \frac{m_i}{M_i}\right)\frac{s_i^2}{m_i}\right] \tag{2.1.23}$$

where $\bar{M}$ is the average PSU size and s_r^2 is the variance of PSU estimated totals.

Ratio estimation can increase the precision of the estimated means and totals as compared to results without ratio estimation. This can result in a more accurate estimate of t_y or $\bar{y}_u$. Ratio estimation will be explored more in the context of sensitivity and specificity, described in the next chapter.

2.2 Introduction to NHANES

The National Health and Nutrition Examination Survey (NHANES) is a US nation-wide study conducted regularly by the Center for Disease Control and Prevention's (CDC) National Center for Health Statistics (NCHS). This survey is conducted to collect information about the health and nutritional status of individuals in all 50 states of the US and D.C., and is used in estimating the prevalence of certain diseases and conditions. Data is collected through three processes: a household screener, an interview, and an examination to collect measurements on desired health-related variables. Due to the possibility that annually released data could increase the chance of breaking anonymity of participants, as well as the lack of reliability in small sample sizes coming from one-year intervals, results from the NHANES data are collected and reported every two years.

The NHANES study is based on a complex survey design consisting of both clustering and stratification elements. The study employs a four-stage clustering design, in which the primary sampling units represent all counties in the entire US. The overall sample was designed to keep an approximately equal sample size per PSU, with a target of 333 examined individuals per each of the 15 locations per year (Johnson et al., 2014). In the case that any PSU had lower than the required minimum sample size, some adjacent counties were combined. A total of 328 counties ended up with lower than the minimum required sample size and were therefore combined with one or more neighboring counties to create larger samples (with the requirement of being within the same state and a maximum of 125 miles distance between counties). From a total of 3100 counties in the US, 2846 PSUs were created to reach desired sample sizes and minimize variance in smaller counties. PSUs in the data collection process were selected with a probability proportional to size (PPS). Thus, larger PSUs had a higher chance of being selected. A sample of 60 PSUs were selected to be studied between 2011-2014, with a random allocation of 15 PSUs per year. The measure of size (MOS) of each PSU was determined beforehand based on attaining desired health estimates including sex, age, race, income, and Hispanic origin. This MOS describes a weighted average of population counts, with higher weights given to PSUs with a greater number of individuals in domains (demographic groups classified by age, sex, race, income, and Hispanic origin) selected for oversampling. PSUs with such a large MOS that they had a probability of 1 were classified as certain PSUs, and had less or more than 24 SSU segments. PSUs with a probability of less than 1, and therefore were not selected with certainty, were classified

as noncertain PSUs, and had 24 segments. From the total of 60 PSUs in the 2011- 2014 surveys, 8 PSUs were deemed as certain. Noncertain PSUs were selected in a way to minimize the number of PSUs that were used in both the 2007-2010 and the 2011-2014 studies. PSUs were also designed in order to minimize the distance a participant would have to travel to reach an examination center, which therefore increases response rates. Selected PSUs were classified as variance units, then becoming masked in order to protect the release of the identity of these PSUs (which might affect participants becoming identified). Data was then clustered by a secondary sampling unit (SSU) that consists of area segments ("census blocks"), which are a group of housing units located next to one another. Segment samples were selected using stratification by examining minority density and geography. The next stage of clustering was the tertiary sampling unit (TSU), which represented dwelling units (DU), such as houses, apartments, and even dormitories. The number of DUs was chosen to create an approximately equal probability sample of DUs in most of the 50 states. To choose these, all dwelling units were listed in each area segment, and a pre-specified sampling rate was set to select the number of listed DUs. DUs were selected with equal probability at a rate equal to the maximum segment rate within segments. Only occupied DUs were eligible for examination. Eight subsamples were then chosen by first selecting a PSU, then SSU, then TSU, then selecting a sample within that group. The goal was to create an approximately equal probability sample between domains. Within each household, all individuals were listed, and the tested individuals were chosen by their demographic traits. The type and number of individuals selected in each household was designed to create self-weighting samples

for each sub-domain (giving each unit the same chance of being selected in the sample), while maximizing the number of sampled individuals per household. The individuals were also selected in a way to reach the desired target sample size for each domain. Each domain was first given a desired sampling rate to reach. Subsample rates were selected in a way to minimize variance in between sampling rates within a stratum, and therefore attaining desired precision. This was done by calculating a sampling rate for each domain, which factors in the target sample size, the expected response rate, and estimated population size. An overall sample size was then calculated to meet each domain target, with some domains requiring oversampling in order to meet expected sample sizes (such as Asian individuals under 12 and males aged 12-19). Oversampling also helps increase reliability and reduce variance while increasing precision of the results.

The NHANES study also uses stratification to increase precision of the estimators. This stratification design focuses primarily on the efficiency of results for the four-year data collection intervals, then secondarily on the efficiency of two-year intervals. The 2011-2014 NHANES data was stratified in state groups, with major and minor strata. Five state groups and one extra state group being solely California were used, each having two or three major strata for a total of 13 major strata. These 13 major strata and 52 minor strata were designed for the 4-year stratification scheme, based on state groupings and urban-rural population distribution of PSUs and on demographics of the PSUs, respectively. The 13 major strata were selected based on their results from different health rankings, including death rate, prevalence of high blood pressure, and percent of adults with poor nutrition. Within each major stratum, there were four minor

strata. The four minor strata were paired within each major stratum by minority concentration, percentage of rural population, and percentage of white or other populations below poverty level. Two-year stratification schemes included a two non-certain PSU per major stratum design, with each PSU being from a different minor stratum. Certain PSUs were not selected within stratum. Only one PSU was chosen from each minor stratum.

2.2.2 Weights

The sampling weight describes the number of target population individuals each participant in the study represents. Sample weights in the 2011-2014 NHANES data were adjusted based on sampling rates, response rates, and coverage rates of domains. The overall goal was to create sample weights that were an accurate, unbiased representation of national estimates. Since there were different response rates for the three processes of data collection (screening, interview, and examination), three different levels of weights were also calculated. Sampling weights were first calculated to compensate for unequal probabilities of selection, then adjusted for non-response to avoid any bias. Finally, weights were post-stratified, a method of adjusting sample weights to compensate for under-coverage or over-coverage of certain demographic groups, to minimize variance in future estimations. These three steps were done for each of the sampling weights between screenings, interviews, and examinations. Final weights were calculated as the product of the weight at base stage, the non-response adjustment, any trimming adjustment, and the poststratification adjustment. Subsampled individuals selected to perform further testing

were chosen in a way to be representative of the target population. For this, new subsample weights were calculated like before.

2.3 Data Preprocessing

This analysis used NHANES data from two cycles, 2011-2012 and 2013-2014. The 2011-2012 and 2013-2014 data sets are comprised of 9,756 and 10,175 participants, respectively.

Data for this research has been downloaded from the NHANES website. The data used in this analysis has been categorized into variables from multiple categories: (1) Demographic Variables and Sample Weights (DEMO), (2) Body Measures (BMX), (3) Blood Pressure & Cholesterol (BPQ), (4) Blood Pressure (BPX), (5) Diabetes (DIQ), (6) Plasma Fasting Glucose (GLU), (7) Cholesterol – HDL (HDL), (8) Current Health Status (HSQ), (9) Medical Conditions (MCQ), (10) Muscle Strength – Grip Test (MGX), (11) Physical Activity (PAQ), and (12) Cholesterol – LDL & Triglycerides (TRIGLY).

To create the sampling design, corresponding variables from the DEMO category were used. The data was stratified by SDMVSTRA, a pseudo-stratum, and clustered by SDMVPSU, a pseudo-primary sampling unit. Weights were defined in the variable WTMEC2YR. The sample target for these sampling design variables were males and females aged 0 to 150 years. To maintain consistency for data between separate individuals, the identifier sequence number SEQN for each participant was retained in DEMO as well as all other datasets.

Although NHANES reports data on a large number of factors, this analysis only examined a subset of those variables. The primary focus of this analysis was participants aged 12-24 years (n=3,759). Subjects were also excluded from analyses if they reported having current infections (i.e., head or chest cold, stomach or intestinal illness, flu, pneumonia, or ear infection) during data collection (n=735). Those with incomplete, missing, or "don't know" cardiometabolic risk (CMR) values, or declined response on covariates (n=1,991) were also excluded from analyses. In total, 1,033 subjects were considered for further analysis.

The sample analyzed includes subjects from all strata and PSUs. Moreover, the proportion of subjects retained for analysis in each PSU and stratum relative to the size 1,033 of the sample analyzed is comparable with the proportion of all NHANES subjects in each PSU and stratum relative to the NHANES total sample size of 19,931. Thus, the sample retained for analysis continues to be representative.

Table 2.1 lists the descriptions, as well as potential values, for all other variables retained. The identifier SEQN number was retained in every data set to ensure correspondence between data from the same individuals. Continuous data was recorded with measurements, and categorical data was recorded as described below. Missing values are denoted with a period in the dataset and were excluded from the analysis. For continuous data, 7777 denotes a refused answer, and 9999 denotes a "Don't know" answer. For categorical variables, a value of 7 denotes a refused response, and a value of 9 denotes a "Don't know" response. Any other specific categorical responses are listed explicitly below.

Table 2.1: Description of Retained Data Set Variables

Data Set	Variable Name	Variable Description	Categorical Potential Values
DEMO	RIDEXAGM[a]	Age in months at time of examination, reported for persons aged 19 or younger	0 to 239
	RIAGENDR	Gender of participant	1= Male 2 = Female
	RIDAGEYR	Age in years at screening	0 to 79 80 = 80 years or age or over
	INDFMPIR	Ratio of family income to poverty	0 to 4.99 5 = Value ≥ 5.00
	RIDRETH3	Race/Hispanic Origin	1 = Mexican American 2 = Other Hispanic 3 = Non-Hispanic White 4 = Non-Hispanic Black 6 = Non-Hispanic Asian 7 = Other Race – Including Multi-Racial
BMX	BMDBMIC[b]	BMI category	1 = Underweight 2 = Normal Weight 3 = Overweight 4 = Obese
	BMXWAIST[c]	Waist circumference (cm)	38.7 to 176 (2011-2012) 40.2 to 177.9 (2013-2014)
	BMXWT	Weight (kg)	3.6 to 216.1 (2011-2012) 3.1 to 222.6 (2013-2014)
	BMXBMI[c]	Body Mass Index (kg/m^2)	12.4 to 82.1 (2011-2012) 12.1 to 82.9 (2013-2014)
BPQ	BPQ020[d]	"Have you ever been told by a doctor or other health professional that you had hypertension, also called high blood pressure?"	1 = Yes 2 = No
	BPQ040A[d]	"Because of your high blood pressure/hypertension, have you ever been told to take prescribed medicine?"	1 = Yes 2 = No

Table 2.1 - Continued

Data Set	Variable Name	Variable Description	Categorical Potential Values
BPX	BPXSY1[e]	1st reading Systolic Blood Pressure (mm Hg)	74 to 238 (2011-2012) 66 to 228 (2013-2014)
	BPXSY2[e]	2nd reading Systolic Blood Pressure (mm Hg)	74 to 234 (2011-2012) 66 to 230 (2013-2014)
	BPXSY3[e]	3rd reading Systolic Blood Pressure (mm Hg)	74 to 232 (2011-2012) 62 to 228 (2013-2014)
	BPXSY4[e]	4th reading (if necessary) Systolic Blood Pressure (mm Hg)	78 to 226 (2011-2012) 80 to 212 (2013-2014)
	BPXDI1[e]	1st reading Diastolic Blood Pressure (mm Hg)	0 to 120 (2011-2012) 0 to 122 (2013-2014)
	BPXDI2[e]	2nd reading Diastolic Blood Pressure (mm Hg)	0 to 134 (2011-2012) 0 to 116 (2013-2014)
	BPXDI3[e]	3rd reading Diastolic Blood Pressure (mm Hg)	0 to 128 (2011-2012) 0 to 118 (2013-2014)
	BPXDI4[e]	4th reading (if necessary) Diastolic Blood Pressure (mm Hg)	0 to 130 (2011-2012) 0 to 128 (2013-2014)
DIQ	DIQ010[f]	"Other than during pregnancy, have you ever been told by a doctor or health professional that you have diabetes or sugar diabetes?"	1 = Yes 2 = No 3 = Borderline
GLU	LBXGLU[g]	Fasting glucose (mg/dL)	39 to 382 (2011-2012) 51 to 421 (2013-2014)
HDL	LBDHDD	Direct HDL Cholesterol (mg/dL)	14 to 175 (2011-2012) 10 to 173 (2013-2014)

Table 2.1 - Continued

Data Set	Variable Name	Variable Description	Categorical Potential Values
HSQ	HSQ500	"Did you have a head cold or chest cold that started during those 30 days?"	1 = Yes 2 = No
	HSQ510	"Did you have a stomach or intestinal illness with vomiting or diarrhea that started during those 30 days?"	1 = Yes 2 = No
	HSQ520	"Did you have flu, pneumonia, or ear infections that started during those 30 days?"	1= Yes 2 = No
MCQ	MCQ035	"Do you still have asthma?"	1= Yes 2 = No
	MCQ010	"Has a doctor or other health professional ever told you that you have asthma?"	1= Yes 2 = No
MGX	MGXH1T1	Grip strength (kg), hand 1, test 1	5 to 79.2 (2011-2012) 4 to 81.5 (2013-2014)
	MGXH1T2	Grip strength (kg), hand 1, test 2	5.2 to 83.7 (2011-2012) 5 to 77.6 (2013-2014)
	MGXH1T3	Grip strength (kg), hand 1, test 3	5 to 85.8 (2011-2012) 5 to 81.2 (2013-2014)
	MGXH2T1	Grip strength (kg), hand 2, test 1	0 to 77.3 (2011-2012) 4 to 79.5 (2013-2014)
	MGXH2T2	Grip strength (kg), hand 2, test 2	5 to 83.8 (2011-2012) 5 to 81.4 (2013-2014)
	MGXH2T3	Grip strength (kg), hand 2, test 3	5.1 to 82.9 (2011-2012) 5.1 to 82.8 (2013-2014)
	MGD130	Dominant hand	1 = Right-handed 2 = Left-handed 3 = Use both hands equally
	MGATHAND	Begin the test with this hand	1 = Right 2 = Left
PAQ	PAD680	Minutes sedentary activity per day	0 to 1380 (2011-2012) 0 to 1200 (2013-2014)
TRIGLY	LBXTR	Triglyceride (mg/dL)	18 to 1562 (2011-2012) 13 to 4233 (2013-2014)

All retained variables were then merged into one dataset using the common identifier SEQN. Next, a new categorical variable ageY was created to subset the ages from RIDAGEYR, in which ages 12-17 were now denoted "15" and 18-24 as "21."
A mean arterial pressure (MAP) value was then calculated. For this, the mean systolic blood pressure (SBP) and mean diastolic blood pressure (DBP) were calculated. Then the new MAP was calculated as such:

$$\text{MAP} = \text{DBP} + \frac{1}{3}(\text{SBP-DBP}) \tag{2.3.1}$$

Next, the maximum handgrip strength was calculated for each hand to determine the dominant hand. To do so, the maximum reading of each hand was calculated (MAXHAND1 = Maximum of hand 1, MAXHAND2= Maximum of hand 2), and then the maximum handgrip strength was determined using MGATHAND and MGD130. For example, if MGATHAND=2, meaning the test began with the left hand, but MGD130=1, meaning the individual was right-handed, then the max hand grip strength was calculated by MAXHAND2, the maximum of hand 2. This was repeated for the other combinations. Individuals who were equally dominant in both hands had a maximum grip strength calculated by the mean of all their handgrip strength readings. Finally, the normalized grip strength was calculated as the HANDVALUE/BMXWT, the handgrip strength value divided by one's weight, both in kg. A composite cardiometabolic z-score was then calculated per individual by summing the normalized values of each cardiometabolic component. To do so, the overall mean and standard deviation were calculated for BMXWAIST, LBXTR, LBHDDN, LBXGLU, and MAP values. Waist circumference was used as a measure of abdominal obesity. Blood measures including triglycerides,

high-density lipoprotein cholesterol (HDL-C), and glucose were included as they are key markers indicative of cardiometabolic health. MAP is generally considered a better indicator of blood perfusion to vital organs (DeMers & Wachs, 2021). All variables were chosen based on components of the metabolic syndrome and current recommendations for CMR factor clustering in adolescents (Brouwer et al., 2013; Magge et al., 2017).

Then each individual value for these components was normalized by subtracting the overall mean and dividing by the standard deviation. Note that the negative version of each HDL value was used, as HDL cholesterol is considered "good" cholesterol, and thus a higher value would only improve one's cardiometabolic status, while a higher value of the other components would negatively impact the cardiometabolic status. Finally, the total composite z-score for each subject was calculated as the sum of each of these normalized components, as such:

$$\text{ZSCORE} = \text{ZWC} + \text{ZTRIG} + \text{ZHDL} + \text{ZGLU} + \text{ZMAP} \qquad (2.3.2)$$

where ZWC is the normalized waist circumference, ZTRIG is the normalized triglyceride value, ZHDL is the normalized negative HDL value, ZGLU is the normalized glucose value, and ZMAP is the normalized MAP value. Any individuals with a missing total z-score were excluded from further analyses to ensure data consistency. Finally, a binary z-score variable ZBIN was defined, where ZBIN = 1 if the ratio of the z-score and the standard deviation of all the z-scores was at least 1, and 0 otherwise (Ramírez-Vélez et al., 2017). This binary variable represents the gold standard throughout this research, classifying participants at risk of cardiometabolic disease. The diagnostic tests developed will be compared against this gold standard to assess their efficiency and accuracy.

2.4 Kriging-Based Optimization

2.4.1 Introduction to Kriging

Kriging is an interpolation or smoothing method for data recorded at spatial locations (e.g. Cressie, 1993). It originates in geostatistics, but has been recently expanded in other areas, such as analysis of computer experiments (Santner et al., 2003). Specifically, in this area kriging produces a fast metamodel that approximates a computationally intensive computer model characterized by an input/output relation. Mathematically, the computer model is represented by an unknown function y depending on inputs defined in an input space. Based on a sample of inputs and corresponding output function values, this method interpolates or smooths y over the input space and gives the best linear unbiased prediction of y at unsampled points. This method will help improve the computational efficiency of model selection and diagnostic test construction using several diagnostic variables, as described later in Chapter Five of this thesis.

Two R packages, DiceKriging and DiceOptim, have been introduced as ways to both model and optimize these computationally intensive functions. Both packages are developed in the frame of DICE (Deep Inside Computer Experiments). DiceKriging and DiceOptim are beneficial in that they are versatile in specification of covariance kernels and trends, provide fast and efficient estimation, and reduce computational redundancy by recycling intermediate calculations and avoiding computational repetition. Details about these two R packages and the statistical background below can be found in Roustant et al. (2012).

2.4.2 Statistical Background

The response variable y is assumed to be a real-valued function based on the d-dimensional input variable $x = (x_1, x_2, \ldots, x_d) \epsilon D$. The set $\boldsymbol{X} = (x^{(1)}, x^{(2)}, \ldots, x^{(n)})$ consists of input points where outputs $\boldsymbol{y} = (y(x^{(1)}), y(x^{(2)}), \ldots, y(x^{(n)}))$ have been evaluated. The purpose of kriging is to predict $y(x)$ for any $x \epsilon D$ based on the set of evaluated y's. One proposed method of linear predictors is Simple Kriging (SK). In SK, $y(x)$ is one realization of a random process

$$Y(x) = \mu(x) + Z(x) \tag{2.4.1}$$

with $\mu(x)$ a known trend function and Z a random process centered at zero and having known covariance kernel C. The best linear unbiased predictor of $Y(x)$ is obtained by choosing λ that minimize the mean squared error

$$MSE(x) = E[\big(Y(x) - \lambda(x)^T Y(X)\big)^2] \tag{2.4.2}$$

For covariance matrix C of $\boldsymbol{y}$, vector $c(x)$ of covariances between $Y(x)$ and $\boldsymbol{y}$, and vector μ of trend values at design points $\boldsymbol{X}$, the SK mean prediction and variance equations are given by

$$m_{SK}(x) = \mu(x) + c(x)^T C^{-1}(\boldsymbol{y}-\mu) \tag{2.4.3}$$

$$s^2_{SK}(x) = C(x,x) - c(x)^T C^{-1} c(x) \tag{2.4.4}$$

in which m_{SK} interpolates the data and s^2_{SK} is non-negative/zero at design points and does not depend on $\boldsymbol{y}$ (Roustant et al., 2012). These equations match the classical conditional distribution results when the process Z is assumed Gaussian. Thus, Y being conditional on $Y(\boldsymbol{X}) = \boldsymbol{y}$ is Gaussian with conditional mean m_{SK} and conditional variance s^2_{SK}.

In cases where the trend is in the form $\mu(x) = \sum_{j=1}^{p} \beta_j f_j(x)$, with fixed basis functions f_j's and unknown real coefficients β_j's, Universal Kriging (UK) derives the best linear predictions of Y from $Y(\boldsymbol{X})$ observations while estimating the vector $\beta = (\beta_1, \beta_2, \dots, \beta_p)^T$. In UK, the mean vector and covariance matrix may not already be known, but a covariance model is assumed. Then for vector of trend functions $f(x)$, this gives $n \times p$ experimental matrix $F = (f(x^{(1)}), \dots, f(x^{(n)}))^T$ and best linear unbiased estimator $\hat{\beta} = (F^T C^{-1} F)^{-1} F^T C^{-1} \boldsymbol{y}$. Then the equations of UK mean prediction and variance are as such:

$$m_{UK}(x) = f(x)\hat{\beta} + c(x)^T C^{-1}(y - F\hat{\beta}) \tag{2.4.5}$$

$$s^2_{UK}(x) = s^2_{SK}(x) + (f(x)^T - c(x)^T C^{-1} F)^T (F^T C^{-1} F)^{-1} (f(x)^T - c(x)^T C^{-1} F) \tag{2.4.6}$$

(Roustant et al., 2012). Similar to the SK method, the UK method interpolates the function values at design points where s^2_{UK} is zero. Although the mean formulas do not change largely between the two, the variance does, as UK has more variability in its prediction from extra terms. In the context of computer experiments the SK and UK mean predictions are called emulators (as they try to emulate the output by predicting it), or metamodels, and the corresponding variances are called the emulator variances.

2.4.3 Covariance Kernels

A covariance kernel is a parametric-form model for the correlation between any two output values. The covariance kernel $C(u,v)$ has an important effect on the kriging model and must be chosen in the set of positive definite kernels. Once a parametric family of kernels has been chosen, its parameters will be estimated by maximizing a likelihood function or minimizing a cross-validation criterion.

A popular class consists of kernels having stationary kernels that are dependent only on the increment *u*-*v*. One-dimension kernels include Gaussian kernels, Matérn kernels, exponential kernels, among others. For higher dimensions, products of 1-dimensional admissible kernels are taken, called separable kernels. For variance $\sigma^2 > 0$ and $h = (h_1, h_2, \ldots, h_d) = u\text{-}v$, based on a 1-dimensional covariance kernel *g*

$$c(h) = C(u,v) = \sigma^2 \prod_{j=1}^{d} g(h_j; \theta_j) \tag{2.4.7}$$

Different choices for the underlying covariance kernels impact the smoothness of the random processes. The more differentiable the kernel, the smoother the realizations of the corresponding random process will be. For example, Gaussian covariance will be very smooth as it has derivatives at all orders, while Matérn covariance will be smoother for $v = 5/2$ (differentiable twice) than $v = 3/2$ (differentiable once). By contrast, the exponential kernel corresponds to Gaussian processes with continuous but nondifferentiable paths. The covariance kernels for each of these distributions are given in Table 2.2 below (Roustant et al., 2012).

Table 2.2: Covariance Kernels of DiceKriging

Distribution	**Covariance Kernel**
Gaussian	$g(h) = \exp\left(-\frac{h^2}{2\theta^2}\right)$
Matérn, $v = 5/2$	$g(h) = \exp\left(1 + \frac{\sqrt{5}\lvert h\rvert}{\theta} + \frac{5h^2}{3\theta^2}\right)\exp\left(-\frac{\sqrt{5}\lvert h\rvert}{\theta}\right)$
Matérn, $v = 3/2$	$g(h) = \exp\left(1 + \frac{\sqrt{3}\lvert h\rvert}{\theta}\right)\exp\left(-\frac{\sqrt{3}\lvert h\rvert}{\theta}\right)$
Exponential	$g(h) = \exp\left(-\frac{\lvert h\rvert}{\theta}\right)$

2.4.4 Expected Improvement (EI) and Efficient Global Optimization (EGO)

An important application of kriging is in the optimization of computationally intensive functions. Using the kriging emulator directly in e.g. the minimization of such a function may lead to erroneous solutions unless the kriging emulator variance is taken into account. This is accomplished by solving a companion optimization problem, namely the maximization of the expected improvement (EI). Details about the relation between the two optimization problems can be found in Jones et al (1998).

The Improvement (I) function at a new candidate optimum point x is defined as $I(x)=(min(y(X)) - y(x))^{+} := max[min(y(X)) - y(x), 0]$. In the case of function minimization, the improvement is the difference between the minimum of function values over the sampled set X and the new function value at x if this difference is positive, and zero otherwise. Since $y(x)$ is not known before the point x has been sampled, $I(x)$ is also unknown. However, its expected value given $Y(X) = \boldsymbol{y}$ can be computed under the Gaussian process model assumption. Denote the N(0,1) cdf Φ and pdf ϕ. Then

$$EI(x) = E\left[min\left(Y(X)\right) - Y(x))^{+} \middle| Y(X) = y\right] \tag{2.4.8}$$

$$= E[min(y) - Y(x))^{+} | Y(X) = y]$$

$$= \left(min(y) - m(x)\right)\Phi\left(\frac{\min(y) - m(x)}{s(x)}\right) + s(x)\phi\left(\frac{min(y) - m(x)}{s(x)}\right)$$

where $m(x)$ is the emulator at point x and $s(x)$ is its standard error. This formula allows for fast evaluations of EI, as well as the calculation of its gradient and higher-order derivatives, which becomes useful in DiceOptim for speeding up EI maximization. This

criterion is null at already visited points x, and positive elsewhere with a magnitude increasing with $s(\cdot)$ and decreasing with $m(\cdot)$.

The efficient global optimization (EGO) algorithm relies on the EI criterion. Beginning with design X, EGO adds new points x in regions where the objective function is close to its optimum (exploitation) or where the emulator variance is large (exploration). The algorithm stops when the improvement in EI does not increase enough to warrant further iterations.

2.4.5 Examples with DiceKriging

The DiceKriging package allows for the design X and trend to be specified, with trends not limited to only polynomials. Then, new location(s) X^{new} for prediction/simulation can be conveniently specified. Either SK or UK can be chosen, as well as a particular covariance structure. As stated earlier, the choice of covariance function impacts the smoothness of the Gaussian process sample functions. Gaussian covariance kernels were the most regular, followed by Matérn (v = 5/2 then v = 3/2) and exponential kernels.

The DiceKriging package was tested on the 2-dimensional Branin-Hoo function, which is often used in global optimization with three global minimizers. For this function, let a, b, c, r, s, and t be constants, and $x = \{x_1, x_2\}$ be the input variables. Then, the analytic form of the Branin-Hoo function is

$$f(x) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1-t)\cos(x_1) + s \qquad (2.4.9)$$

A 4x4 grid of inputs was then used to construct a kriging emulator. To this end, a first-order polynomial was used for the trend and a Gaussian covariance kernel for the random process, under the assumption that the Branin-Hoo function is smooth. Then the kriging trend and covariance were estimated using the km function in R. Next, both the function (left panel) and its emulator (middle panel) were plotted in Figure 2.4.1 to verify for a similar result. In this figure, the left plot shows the contour plot of the Branin-Hoo Function, the middle plot shows the mean of the Ordinary Kriging metamodel, and the right plot shows the variance of the Ordinary Kriging metamodel.

It can be seen that the emulator approximates the function well, and adding more test points to the grid would result in an even better match. The kriging variance contour plot was also plotted (right panel) showing that it is zero at the design points. Finally, a leave-out-one cross-validation method for the Kriging method on the Branin-Hoo function is shown in Figure 2.4.2, demonstrating further the emulator accuracy.

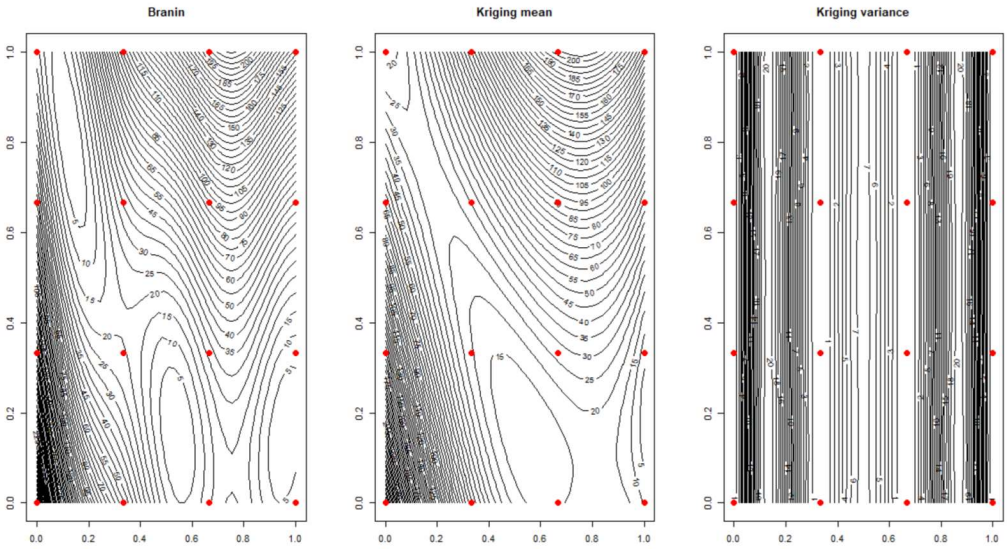

Figure 2.4.1: Branin-Hoo Function and Emulator

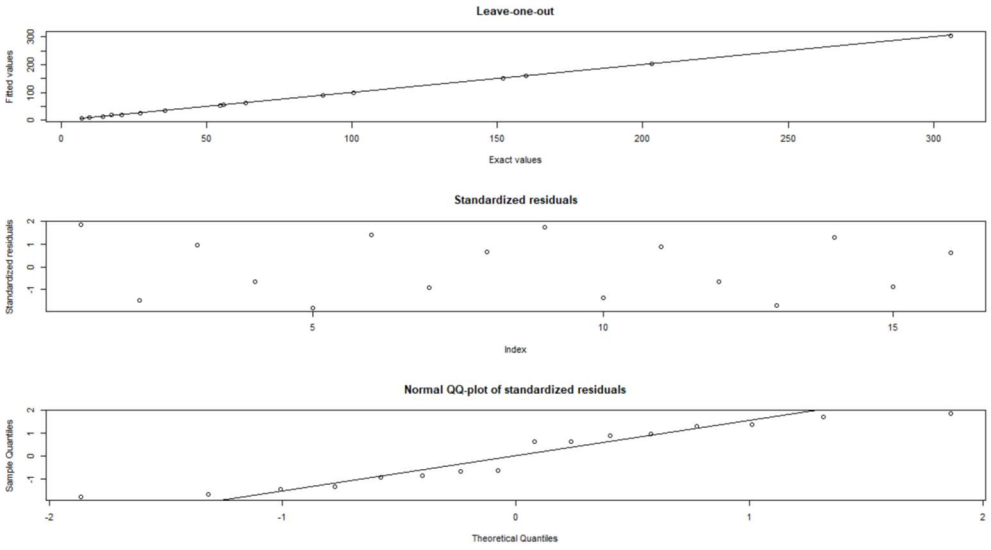

Figure 2.4.2: Validation method for kriging on the Branin-Hoo Function

From Figure 2.4.2, the linear trend in the leave-one-out plot shows agreement between the exact and fitted values. The standardized residuals plot looks for any random noise and showed that there were no outliers., Finally, the normal QQ plot showed that the data was fairly normally distributed. Thus, the kriging emulator was fairly accurate in predicting the missing point.

2.4.6 Examples with DiceOptim

The main focus of DiceOptim is the use of the Expected Improvement (EI) to optimize functions, which evaluates a given function at a point where the improvement is expected to be maximized. This package allows for computation of the kriging mean predictor and variance, prediction intervals, and EI for x, while allowing for an adjustable

rectangular search domain, starting point, and specified control parameters. This package was tested using a 2-dimensional Branin function with an initial 15-point Latin Hypercube (LH) design. The EI method requires a kriging emulator that needed to be estimated. The selection of the initial design may influence the EI results, unless enough points are added by the EGO algorithm to ensure the global optimum is identified. In Figure 2.4.3, the blue triangles represent the initial optimal LH design, and the red points represent the visited points during the EGO search. The contours in the left panel represent the Branin function showing that all three optima were identified, while the right panel represents the EI function at the last iteration of the EGO algorithm.

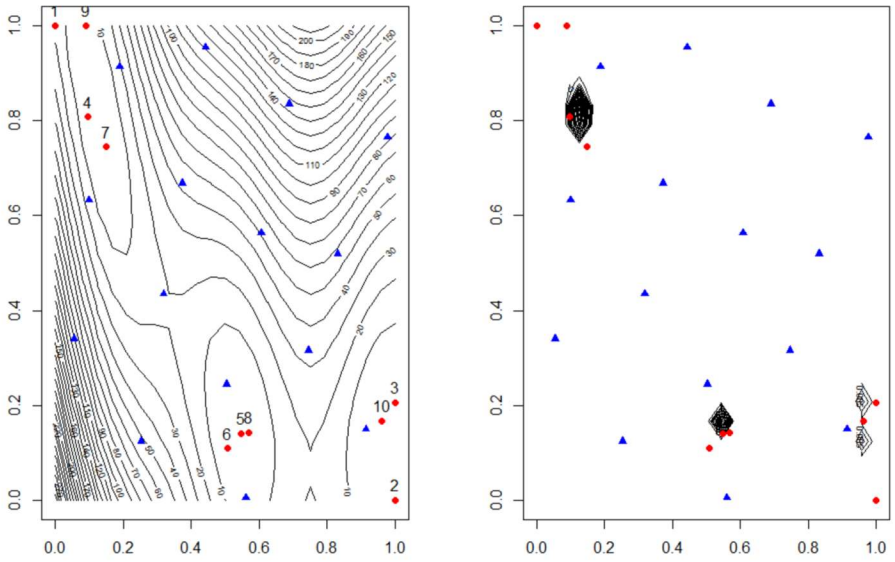

Figure 2.4.3: Optimizing the Branin function using EI and EGO Algorithm

CHAPTER THREE

ROC CURVES FOR SURVEY DATA

3.1 Introduction

Classifying subjects into one of two categories (diseased or non-diseased) is beneficial in establishing a subject's status to properly assess treatment plans and disease managements. However, when using a diagnostic variable there is generally not a clear-cut distinction between the two categories, so an optimal cut-point for the diagnostic variable needs to be established. Ultimately, the goal is to find an optimal cut-point at which the true positive rate (sensitivity) and true negative rate (specificity) are simultaneously maximized. In this chapter, receiver operating characteristic (ROC) curves will be explored as a method of finding this optimal cut-point to serve as a diagnostic tool. This method uses the sensitivity and specificity levels to create appropriate indices that can find the optimal cut-point.

Currently, the majority of ROC curve methodology has been developed under simple random sample designs. ROC curves have not previously been well-explored under complex survey designs; therefore this chapter will delve into the application of ROC curves with survey data and attempt to implement ROC curves to find optimal cut-points in the NHANES study. Another limitation of ROC curves is the lack of methodology for the use of covariates along with survey data to determine optimal cut-points. Yet another limitation of ROC curves under complex survey designs is that the diagnostic test can only be built using one diagnostic variable. Some of these limitations will be addressed in this chapter, while others are better addressed using new methods as

proposed in the next chapters. The lack of extensive literature on ROCs for survey data highlights both the complexity of the classification problem as well as the need for practical solutions.

3.2 ROC Curves

3.2.1 Review of ROC Methodology

Receiver operating characteristic curves are used to determine diagnostic tests and assess their accuracy. The true disease status is determined by a gold standard, against which the diagnostic tests are compared. Ideally, the sensitivity and specificity are both 100%. However, this generally is impossible in practice, and a diagnostic test is desired that makes the sensitivity and specificity as large as possible. At the other extreme, a useless diagnostic test represents random chance, unable to distinguish between diseased and non-diseased subjects (Akobeng, 2007), for which the number of true positives and false positives are equal.

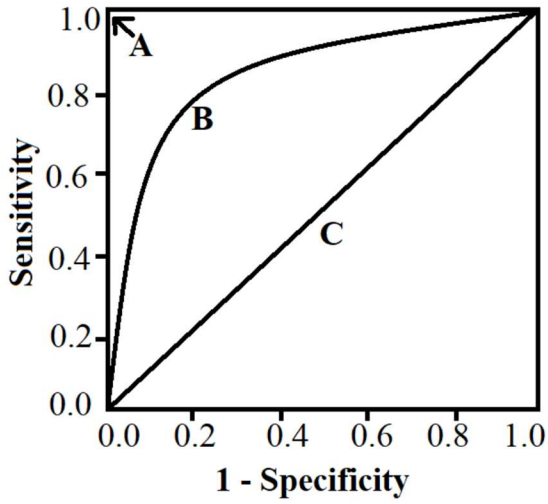

Figure 3.2.1: ROC Curve Standard Examples

In plotting an ROC curve, the y-axis represents sensitivity, and the x-axis represents 1-specificity, as shown in Figure 3.2.1. Generally, the ROC curve will fall somewhere in between the gold standard and random chance line. This ROC curve is calculated for varying candidate cut-points. Line A, being the segments going from (0,0) to (0,1) and (0,1) to (1, 1), represents a diagnostic test as good as the gold standard (in which sensitivity is maximized and 1-specificity is minimized). Line C, being the diagonal line from (0,0) to (1,1), represents random chance (in which there is no distinction between those with disease and those without). Line B is a typical example of an ROC curve, which falls somewhere in between the gold standard and random chance reference lines (Zou et al., 2007).

Figure 3.2.2 shows a nonparametric estimate of a ROC curve, graphing sensitivity versus 1-specificity at varying levels of the cut-point. This method allows for no assumptions to be made, including that of any underlying distributions and it typically creates a jagged curve. Alternatively, a ROC curve can be generated by a parametric model, such as the binormal model. This model assigns the measurements variables to independent normal distributions (with different means and standard deviations corresponding to diseased or non-diseased). This model allows for covariates to be incorporated easily. Optionally, a transformation can be applied (such as a log transformation) in a parametric model, which results in a smoother curve that would better fit the data. This is illustrated in Figure 3.2.2, with the jagged curve representing the nonparametric method, while a parametric method would result in a smooth curve like that of Figure 3.2.1. While the parametric models allow for the inclusion of

covariates, this framework still uses simple random samples, and it is not clear or obvious how the model can be extended to complex survey samples. The method developed in the next chapters allows for the inclusion of covariates in the construction of diagnostic tests and it is applicable to complex survey samples.

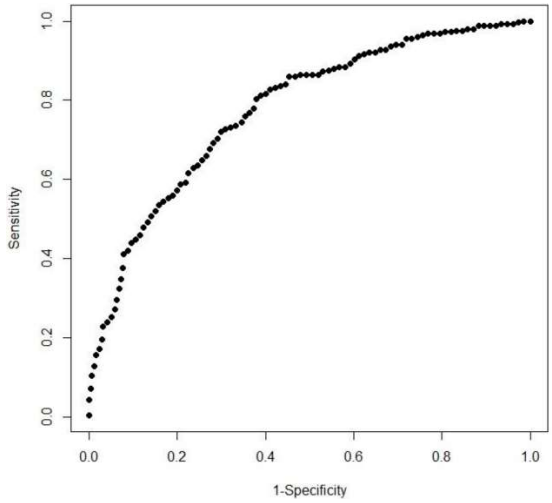

Figure 3.2.2: Nonparametric Method Example

The ROC curve is used to assess whether a diagnostic test can distinguish between diseased and non-diseased subjects (Akobeng, 2007), by allowing for a visual representation of changing sensitivity and 1-specificity rates at varying levels of cut points. In total, the ROC curve is beneficial for three purposes: (1) to determine the optimal cut-point at which sensitivity is maximized and 1-specificity is minimized, (2) to assess the accuracy of disease diagnoses, and (3) to compare two or more diagnostic tests.

In terms of purpose (1), the ROC curve is used to identify an optimal cut-point. The first method of doing so is using the upper left (UL) index, which chooses the point

on the curve closest to (0,1) (Koyama et al., 2016). This point maximizes specificity and sensitivity, and minimizes the following quantity:

$$(1\text{-sensitivity})^2 + (1\text{-specificity})^2 \quad (3.2.1)$$

This cut-point would best differentiate between those with and without disease. The second method of choosing an optimal cut-point is by calculating the Youden Index *J.* This value represents the maximum vertical distance between the ROC curve and diagonal random chance line, and maximizes the following quantity:

$$sensitivity + specificity - 1 \quad (3.2.2)$$

Once this is maximized, the corresponding cut-off point is deemed the best cut-point. These indices have been used mostly in the context of simple random samples. This will be extended to complex surveys later in this chapter.

To quantify the usefulness of the ROC curve, the area under the curve (AUC) is used. The AUC averages the diagnostic accuracy across varying cut-point levels (Zou et al., 2007), and summarizes the quality of the diagnostic test (Akobeng, 2007). The AUC under the gold standard line is 1.0 (deemed "perfect"), and the AUC under the random chance line is 0.5 (deemed "useless"), with the AUC of a realistic ROC curve lying somewhere within 0.5-1.0. More specifically, an AUC of 0.9-1 represents high accuracy, 0.7-0.9 represents moderate accuracy, 0.5-0.7 represents low accuracy, and less than 0.5 being worse than chance (the AUC point estimates can also be accompanied by measures of error or uncertainty). The higher the AUC for the ROC curve, the better the test is at distinguishing disease status.

An ROC curve can be used to compare the accuracy of diagnoses between multiple tests and determine which of them is best. As pointed out by Akobeng (2007), an optimal ROC curve connects points with high vertical coordinates (i.e. high sensitivity) and low horizontal coordinates (i.e. high sensitivity). In practice, it is best to select an ROC curve that maximizes sensitivity if it is important to not miss a diagnosis, since catching a disease early on might give one a better chance of combating it. Conversely, it is best to select an ROC curve that maximizes specificity when the consequences of a false positive diagnosis are serious (Akobeng, 2007).

There has been previous literature that examined using ROC curves for more than a single diagnostic variable. For example, Ferri et al. (2003) explored multi-class ROC curves, where the curve was transformed to a multi-dimensional surface. For this, the area under the curve was extended to the volume under the surface as a verification method of accuracy. However, multi-class ROC studies have not been applied to survey data.

Thus, an ROC curve is used to analyze the varying sensitivity and specificity rates at differing cut-points to choose an optimal cut-point that would maximize both the sensitivity and the specificity. Once an optimal cut- point is selected, the ROC curve can be used to assess the diagnostic accuracy and compare tests and predictive models.

3.2.2 AUC Estimation Applications Under Survey Designs

Although ROC curves are often beneficial in analyzing a diagnostic test, most parametric and nonparametric methods of estimating the AUC are typically not

applicable for complex sample designs. In general, AUC estimation methods are catered towards simple random samples, and not complex survey designs. One of the few exceptions is Yao et al. (2014), who modified a nonparametric estimation method of AUC's to incorporate the complex survey design and the survey weights, resulting in more accurate estimators of the population AUC and their variances.

Yao et al. (2014) applied their method to survey data from the Hispanic Health and Nutrition Examination Survey (HHANES), which has a complex survey design involving both multistage clustering and stratification. For simplification, Yao et al. examined AUC estimation methods for a stratified two-stage cluster sampling (STSCS). Such a simplified survey design may be appropriate when additional levels of clustering do not contribute much to the overall design. Deriving AUC estimators in such a simplified design, however, may be a limitation in other complex surveys where multiple stages of clustering contribute significantly to the overall design.

As an application, they compared three predictors: self-reported body mass index (BMI), measured triceps skinfold, and measured subscapular skinfold, to determine their efficacy in classifying subjects as overweight/obese. This was done by pulling data specifically from Mexican Americans aged 2 to 18 years, with unequal probabilities of selection. When taking into account the sample weights, the self-reported BMI had the highest AUC estimate between the three predictors of obesity. However, without taking into account the sample weights, the AUC for subscapular skinfold was slightly higher than the BMI, with the triceps skinfold having the lowest AUC. The weighted jackknife standard errors were slightly larger than the unweighted. In this example, however, the

results were close between the weighted and unweighted estimators, thus showing that sample weighting was not very informative. Using the weighted estimation, however, did not significantly decrease the precision.

ROC curves were made for each of the three predictors and showed that the subscapular skinfold measurement was better at predicting overweight/obesity status than the triceps skinfold measurement. Although the self-reported BMI had a higher AUC, it was not always the better predictor. BMI was a better predictor when sensitivity was above 50% and specificity was above 70%.

When the study was repeated using subsamples of men and women aged 20-74 years, all three predictors resulted in better classification in females than men, with higher AUCs and smaller variances. Here, self-reported BMI was the best predictor for classifying overweight/obese status. By constructing unbiased estimates of the AUC, Yao et al. were able to incorporate the use of ROC curves to estimate AUC levels (as well as their variances), when using a complex sampling design consisting of both stratification and clustering.

Delving further into using ROC curves with a complex survey design, Sukhatme and Beam (1994) focused on applying stratification to ROC analysis, which allows for the account of extraneous factors. Since stratification divides subjects into different strata, there exists ROC curves within each stratum. These ROC curves can either be analyzed per stratum, or for the population as a whole. In analyzing stratum-specific ROC curves, their goal was to assess that the test performance was improved when accounting for stratification.

To complete this assessment, Sukhatme and Beam (1994) tested a few case-scenarios. One scenario examined the ability of carcinoembryonic antigen (CEA) to detect breast cancer. Focusing on women who, at the time, had evidence of breast cancer, the controls were split into two strata: (1) "healthy" women with no history of cancer or chronic disease, nor evidence of breast cancer at the time, (2) women with benign breast conditions. The cases, however, were not stratified, and thus had to undergo post-stratification. When analyzing the ROC curves, it was found that the AUC variance estimate was smaller than if the control group had not been stratified. It was found to be no significant difference in the ability of CEA to diagnose breast cancer between the two strata of the control groups.

3.2.3 Youden Index and Upper Left Index

The application of ROC curves to assess diagnostic accuracy has been diverse. Koyama et al. (2016) used ROC curves to define the cut-point for two serum biomarkers (serum aminotransferase (AST) and alanine aminotransferase (ALT)) to predict the presence of liver injury and therefore determine whether undergoing a CT scan would be necessary, as it comes with the risk of causing malignancies. Both the upper left and Youden indices were used to determine the optimal cut-point values for these biomarkers. With each of these methods, the sensitivity and specificity, along with the AUC, were then calculated. AST and ALT levels of $\geq$ 109 U/I and $\geq$ 97 U/I, respectively, were found to be the optimal cut-points to detect the presence of liver injury. These cut-point values were both associated with an AUC of 0.88, with AST having UL index of 0.26 and

Youden index of 0.63, and ALT having UL index of 0.25 and Youden index of 0.67 (Koyama et al., 2016). The cut-point values were a little different than similar previous ROC studies. One study by Tian et al. (2012) defined AST ≥ 113 U/I and ALT ≥ 57 U/I to be the optimal cut-point values, another by Tan et al. (2009) found cut-point levels of AST ≥ 83 U/I and ALT ≥ 64 U/I, and a third by Lee at al. found AST ≥ 100 U/I and ALT ≥ 80 U/I. Establishing specific and proper cut-point levels would be particularly useful in determining whether patients with potential blunt liver injury would need to undergo a CT scan.

ROC analysis has also previously been used in studying the association between handgrip strength as a proxy measure for total muscular strength (MS) and metabolic risk. Handgrip strength is an effective tool in assessing MS as it is simple and low-cost. Generally, low MS or handgrip strength is correlated with increased disease and mortality, as low MS contributes to increased cardiometabolic risk factors.

Ramírez-Vélez et al. (2017) studied the association between MS and overall healthiness with metabolic risk in Colombian children aged 9 to 12.9 years and adolescents aged 13.0 to 17.9 years, by attempting to find the optimal MS cut-point to categorize cardiometabolic risk. After collecting data on many metabolic factors, a cardiometabolic risk score (CMR) was calculated by summing the age and sex standardized scores of waist circumference (WC), triglycerides, HDL-cholesterol[1], glucose, and systolic (SBP) and diastolic blood pressure (DBP) levels. The higher the

[1] Multiplied by -1 as HDL is inversely related with cardiovascular risk (Ramírez-Vélez et al.)

CMR, the higher the cardiometabolic risk. In addition, an age-adjusted continuous cardiometabolic risk composite z-score was calculated as such

$$\textit{Composite z-score} = \textit{z-WC} + \textit{z-triglycerides} + \textit{z-HDL-c} + \textit{z-glucose} + \textit{z-SBP} + \textit{z-DBP} \quad (3.2.3)$$

This score was calculated for each participant, in which a CMR score larger than the standard deviation of all CMR scores in the sample classifies the subject at risk of cardiometabolic diseases.

The diagnostic accuracy of the handgrip strength per body mass was analyzed to determine the difference between low and high CMR. Then, an ROC curve was used in which cut-point values were determined using the UL index, and the AUC was calculated. The cut-point level for handgrip strength (kg) / body mass (kg) in children aged 9 to 12.9 years was found to be 0.359 in girls and 0.376 in boys, as compared to 0.440 in girls and 0.447 in boys in adolescents aged 13 to 17.9 years. These were associated with AUC levels of 0.83, 0.84, 0.79, and 0.88, respectively, as well as respective Youden indices of 0.61, 0.62, 0.48, and 0.56 (Ramírez-Vélez et al., 2017). These cut-point values were in agreement with those found by another study by Peterson et al. (2016), who defined high cardiometabolic risk cut-point values to be 0.33 in boys and 0.28 in girls, and intermediate cardiometabolic risk cut-point values to be between 0.33 and 0.45 for boys and between 0.28 and 0.36 for girls.

Ramírez-Vélez et al. (2017) determined that their ROC analyses identified accurately the CMR status across gender and age groups. The cut-points were used to categorize participants into either low or high cardiometabolic risk. It was found that MS

cut-points signified a difference in cardiometabolic risk factors on the basis of handgrip strength relative to body weight in these Colombian children and adolescents, with an inverse relationship between the two. Thus, promoting increased muscular strength may be an effective tool to decrease cardiovascular disease risk.

Note that in the studies above, the indices have been used for ROC curves estimated from simple random samples. Later in this chapter, using these indices for ROC curves based on complex survey samples will be shown.

3.2.4 Covariates and Complex Survey Designs

Besides variables of primary interest, a complex survey sample may also include variables that are not the main focus of the study. However, their inclusion as covariates in the analysis may improve the results related to primary variables and reduce the variance of estimators of interest. Janes et al. (2009) expressed the ROC curve in terms of the cumulative distribution function of percentile values of the diagnostic (or marker) variable, which in further is expressed in terms of covariates. They explored three methods of implementing covariate information into ROC curves. The first method adjusts for covariates (or factors) that affect marker observations among controls. In the second method, the ROC curve is expressed as a function of covariates that affect discrimination. For two normal curves (diseased and nondiseased) with far means, the distinction is very different with good discrimination. In the presence of an overlap, however, the diagnostic variable may not be strong enough, requiring the need of a cut-point threshold. In the third method, information about marker and covariates

contributing to discrimination is combined to assess the marker's additional contribution to discrimination. Although Janes et al. focused only on ROC curves that adjust for covariates under a simple random sample design, the possibility of applying this technique to complex surveys is yet to be assessed and may be a potential topic for future work.

3.2.5 Ratio estimation and its application to ROC curves for survey data

The main contribution of this chapter is to determine optimal cut-points for ROC curves using complex survey data. This is achieved using ratio estimation for sensitivity and specificity levels, followed by the optimization of the Youden and upper left indices as functions of sensitivity and specificity to determine the best cut-point.

Under some complex survey design with both clustering and stratification, let h be the stratum number for a total of H strata ($h = 1, 2, \dots, H$), i be the cluster number within stratum h with a total of n_h sample clusters in stratum h ($i = 1, 2, \dots, n_h$), and j be the unit number within cluster i of stratum h for a total of m_{hi} sample units from cluster i of stratum h ($j = 1, 2, \dots, m_{hi}$). Let n be the total number of observations in the sample. Denote W_{hij} as the sampling weight of unit j in cluster i of stratum h and f_h as the first-stage sampling rate (i.e. the fraction of PSUs selected for the sample) for stratum h. Then let the indicator values $\delta_{hij}(r, c)$ denote whether an observation belongs to cell (r, c) for row r and column c of a two-way table ($r = 1, 2, \dots, R$; $c = 1, 2, \dots, C$), such that

$$\delta_{hij}(r, c) = \begin{cases} 1 \text{ if observation } (hij) \text{ is in cell } (r, c) \\ 0 \text{ otherwise} \end{cases} \tag{3.2.4}$$

Also, let the indicator values $\delta_{hi}(\cdot\, c)$ be defined as such

$$\delta_{hij}(\cdot\, c) = \begin{cases} 1 \text{ if observation } (hij) \text{ is in column } c \\ 0 \text{ otherwise} \end{cases} \tag{3.2.5}$$

Then the ratio estimate can estimate the column proportions as the ratio of cell (r, c) to the total for column c as

$$\hat{P}_{rc}^{c} = \frac{\hat{N}_{rc}}{\hat{N}_{.c}} = \left(\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r, c) W_{hij} \right) / \left(\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(\cdot\, c) W_{hij} \right) \tag{3.2.6}$$

The variance of the column proportions can be estimated using Taylor series linearization, given by

$$\widehat{Var}\left(\hat{P}_{rc}^{c}\right) = \sum_{h=1}^{H} \widehat{Var}_h(\hat{P}_{rc}^{c}) \tag{3.2.7}$$

where if $n_h > 1$,

$$\widehat{Var}_h\left(\hat{P}_{rc}^{c}\right) = \frac{n_h(1 - f_h)}{n_h - 1} \sum_{i=1}^{n_h} (g_{rc}^{hi} - \bar{g}_{rc}^{h})^2 \tag{3.2.8}$$

$$g_{rc}^{hi} = \sum_{j=1}^{m_{hi}} \left(\delta_{hij}(r, c) - \hat{P}_{rc}^{c} \delta_{hi}\ (\cdot\, c) \right) W_{hij}) \,/\, \hat{N}_{.c} \tag{3.2.9}$$

$$\bar{g}_{rc}^{h} = \left(\sum_{i=1}^{n_h} g_{rc}^{hi} \right) / n_h \tag{3.2.10}$$

Additional details can be found in SAS Institute Inc. (2020) and the references therein.

The ratio estimation is implemented in ROC curves through the sensitivity and specificity, since they are both defined as ratios (e.g. Fleiss et al., 2003). Let a 2x2 contingency table be comprised of rows representing a positive or negative diagnosis test result, and columns representing diseased or non-diseased. Then sensitivity (true positive

rate) is the ratio of the total in cell (1,1) to the total in column 1, with the estimate given by

$$SN = \frac{\widehat{N}_{11}}{\widehat{N}_{.1}} \tag{3.2.11}$$

and specificity (true negative rate) is the ratio of the total in cell (2,2) to the total in column 2, with the estimate given by

$$SP = \frac{\widehat{N}_{22}}{\widehat{N}_{.2}} \tag{3.2.12}$$

For both sensitivity and specificity, the variances are estimated using Taylor series variance estimation, as described above.

Once the sensitivity and specificity values have been obtained, the estimates for the Youden index and upper left index can be obtained through the formulas given in section 3.2.1. These estimates, as well as their variances, can be estimated through a replication method for survey data. With the availability of cheap computing power, using replication methods to estimate variance in complex survey designs have become popular due to their simplicity and generality. In particular, for functions of sensitivity and specificity, such as the Youden index and upper left index, Taylor linearization methods are more difficult to apply due to the derivation of the analytical expressions involved, while replication-based variance estimation appears to be an easily implementable alternative.

For a complex survey design with both clustering and stratification, the bootstrap samples are drawn from the original sample in groups, using the same survey design and adjusting the survey weights. For variance estimation using a bootstrap replication

method, let R be the total number of bootstrap samples (i.e. replicates) and let α_r be the replicate coefficient for the r^{th} replicate ($r = 1, 2, \dots, R$). By default, $\alpha_r = 1/R$. To estimate the variance, let $\hat{\theta}$ be the estimator of a parameter vector θ using the full sample and let $\hat{\theta}_r$ be the estimator of θ when the r^{th} bootstrap sample is used. Then the estimated covariance matrix of $\hat{\theta}$ is given by

$$\hat{V}(\hat{\theta}) = \sum_{r=1}^{R} \alpha_r (\hat{\theta}_r - \hat{\theta})(\hat{\theta}_r - \hat{\theta})' \tag{3.2.13}$$

Inference using this estimated covariance matrix is performed, where the number of degrees of freedom is equal to the number of clusters minus the number of strata.

When a jackknife method is used for variance estimation, PSUs are deleted one-at-a-time from the full sample to create jackknife samples (or replicates). In contrast to a bootstrap sample, where samples are taken by leaving group out at a time, leaving one out at a time ensures every combination is taken but results in less estimators. Let R, the total number of replicates, be equal to the total number of PSUs. For each replicate, the sample weights of the remaining PSUs are modified into replicate weights (e.g. SAS Institute Inc. 2020). Similar to the bootstrap method, let $\hat{\theta}$ be the estimator of a parameter vector θ from the full sample and let $\hat{\theta}_r$ be the estimator of θ from the r^{th} replicate when replicate weights are used. Then the estimated covariance matrix of $\hat{\theta}$ is given by

$$\hat{V}(\hat{\theta}) = \sum_{r=1}^{R} \alpha_r (\hat{\theta}_r - \hat{\theta})(\hat{\theta}_r - \hat{\theta})' \tag{3.2.14}$$

As with the bootstrap method, the number of degrees of freedom for the jackknife method is equal to the number of clusters minus the number of strata. The jackknife replication method can be extended to delete a group of PSUs at a time.

3.3 NHANES Results

ROC curves were considered for three variables (NGS, BMXBMI, and RIDAGEYR), each plotted on a set of candidate cut-points to determine their optimal cut-point. These variables were chosen as diagnostic variables since they were statistically significant in a logistic regression model, with more details to be given in Chapter Four. To do so, a variable was dichotomized at each candidate cut-point. NGS was dichotomized on a range of 0.30 to 0.45 in increments of 0.01, BMI on a range of 17 to 37 by increments of 1, and AGE on a range of 12 to 24 by increments of 1. Then the sensitivity and specificity values were calculated with respect to the gold standard binary CMR z-score and the diagnostic binary variable dichotomized at each potential cut-point. To calculate these estimates, the R function svyratio was used, which implements ratio estimation under a complex survey design, and Taylor linearization to calculate standard errors. Sensitivity was calculated as the ratio of subjects that had both a binary CMR z-score of 1 and a dichotomized diagnostic variable of 1, over all subjects with a binary CMR z-score of 1. Specificity was calculated as the ratio of those that had both a binary CMR z-score of 0 and a dichotomized diagnostic variable of 0, over all subjects with a binary CMR z-score of 0. The sensitivity and specificity values were computed for each potential cut-point, then the 1-specificity values were calculated and plotted against the

sensitivity values to obtain the ROC curves, shown in Figure 3.3.1 below. Note that the extreme coordinates of (0,0) and (1,1) were also plotted and used in further calculations as reference end points.

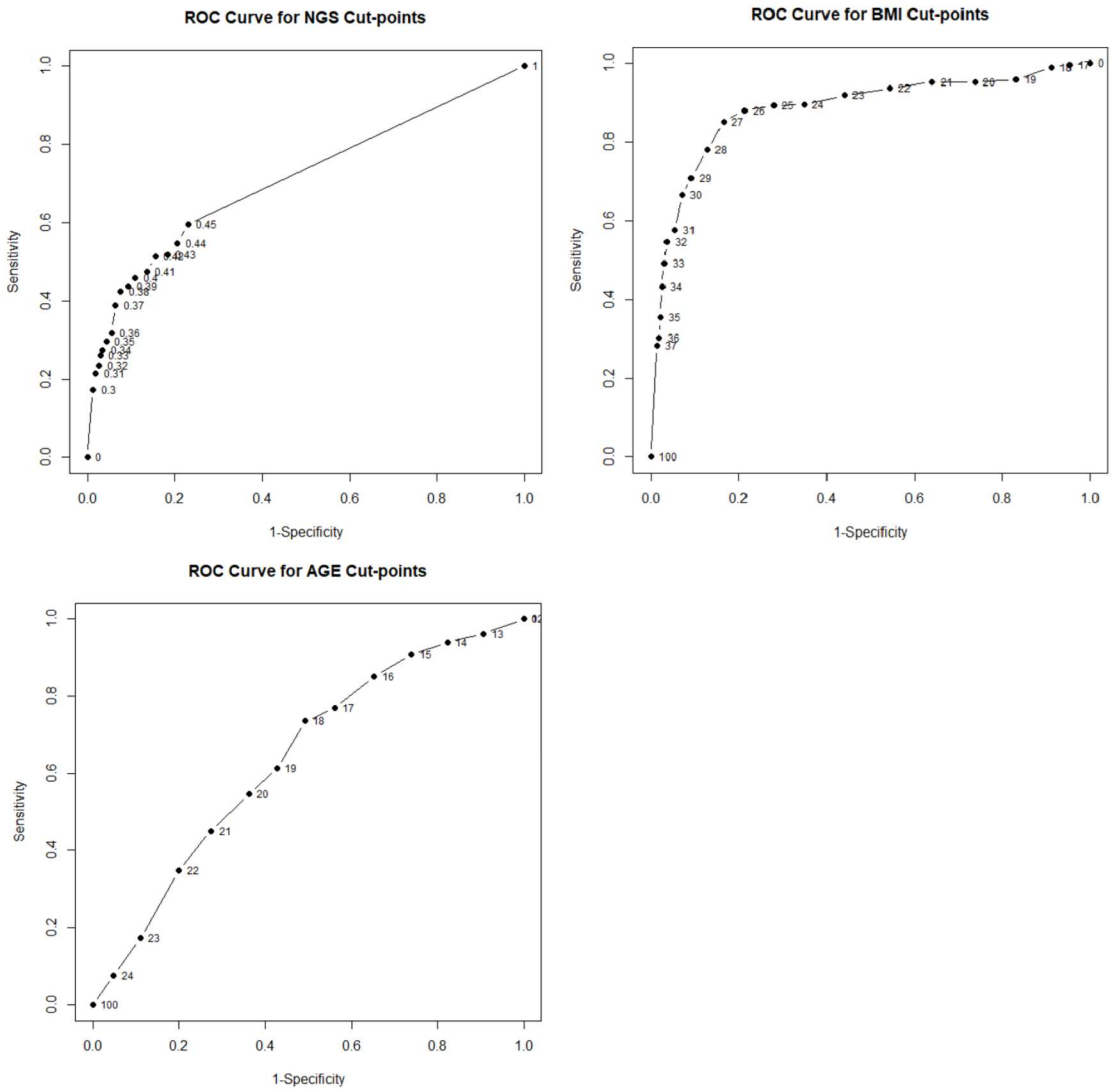

Figure 3.3.1: ROC Curve for 1-Dimensional Cut-Points

Once the ROC curves were plotted, the approximate area under the curve (AUC) was calculated for each graph. Given the limitations of AUC from complex survey designs, here a different approach for the computation of the AUC was explored based on

Riemann sum approximation of an integral. A Riemann sum divides the area under the curve into rectangles, calculating the total area under the curve as the sum of the areas of the individual rectangles. The width of each rectangle is calculated as the difference between x-values, and the length is calculated as some measurement of the y-values. The choice of the y-value is subjective, as differentiating between the left Riemann sum (using the lower of the two y-values in each rectangle) and the right Riemann sum (using the higher of the two y-values in each rectangle) will affect the total area. For this calculation, a trapezoidal Riemann sum was used with the trapz R function from the pracma package. This method uses the average of the two y-values in each rectangle as the length rather than selecting one of the y-values. The absolute value of the difference between the right and left Riemann sums was also calculated as a bound for the integral approximation error. Table 3.1 lists the trapezoidal Riemann sum AUC estimates as well as the error bounds for each of the three variables.

Table 3.1: 1-Dimensional Trapezoidal Riemann Sum AUC Approximations

Variable	**AUC Approximation**	**Error Bound**
NGS	0.7116769	0.3160465
BMI	0.8840858	0.02073914
AGE	0.6384293	0.06682345

The higher the AUC, the more accurate a diagnostic test is. NGS and BMI had approximate AUC's in the range of 0.7-0.9, representing moderate accuracy. AGE had an approximate AUC in the range of 0.5-0.7, representing low accuracy. However, NGS had a much higher integral approximation error bound than BMI and AGE. Thus, this 1-

dimensional method using NGS may not be highly accurate in distinguishing disease status. Overall, the BMI appears to be the best diagnostic variable as it has the highest AUC and the smallest approximation error.

Table 3.2: 1-Dimensional ROC Curve Indices Optimal Cut-Points

Variable	Minimum Upper Left Index	Upper Left Index Min Cut-Point	Maximum Youden Index	Youden Index Max Cut-point
NGS	0.2158827	0.45	0.3661154	0.45
BMI	0.04922838	27	0.686686	27
AGE	0.3121351	18	0.2435	18

Next, the Youden Index and Upper Left Index values were calculated for each variable to find the optimal cut-point. These point estimates of both indices were first calculated by taking the estimates of the sensitivity and specificity as the ratios of true positives and true negatives for each potential cut-point in the range, and using the Youden index or upper left index equations to find the index value at each point. Then the minimum upper-left index and maximum Youden index value was obtained for each variable. The results for this search are listed in Table 3.2. For each of the three variables, the optimal cut-point was the same between the two indices. Thus the optimal cut-point was 0.45 for NGS, 27 for BMI, and 18 for AGE using ROC curves.

Table 3.3: 1-Dimensional Ratio Estimation with Taylor Linearization versus Jackknife Replication Method

Variable	Taylor Linearization Sensitivity Estimate	Jackknife Replication Sensitivity Estimate	Taylor Linearization Specificity Estimate	Jackknife Replication Specificity Estimate
NGS	0.5965191 (0.05315409)	0.59652 (0.0534)	0.7695963 (0.01783795)	0.7696 (0.0178)

Table 3.3 – Continued

Variable	Taylor Linearization Sensitivity Estimate	Jackknife Replication Sensitivity Estimate	Taylor Linearization Specificity Estimate	Jackknife Replication Specificity Estimate
BMI	0.8518702 (0.03227577)	0.85187 (0.0324)	0.8348155 (0.01861215)	0.83482 (0.0186)
AGE	0.7357438 (0.03043661)	0.74574 (0.0305)	0.5077564 (0.02587617)	0.50776 (0.0259)

The sensitivity and specificity at each of these optimal cut-points were estimated using both the Taylor linearization method and the jackknife replication method. The point estimates along with their standard errors are displayed in Table 3.3, where it can be seen that both methods give similar results. This similarity verifies the accuracy of the replication method in estimating the sensitivity and specificity values, which can further be used to obtain jackknife-based point estimates and standard errors for both the Youden and upper left indices, listed in Table 3.4.

Table 3.4: 1-Dimensional Replication Method Youden and Upper Left Index Estimates

Variable	Youden Index Point Estimate	Youden Index Standard Error	Upper Left Index Point Estimate	Upper Left Index Standard Error
NGS	0.36612	0.0586	0.21588	0.0453
BMI	0.68669	0.0357	0.049228	0.0108
AGE	0.2435	0.0394	0.31214	0.0297

These point estimates were similar to those in Table 3.2 found by direct calculation of both indices for all three diagnostic variables.

Next, the diagnostic tests defined by these optimal cut-points were used to estimate the proportions of subjects at risk of cardiometabolic disease, and their confidence intervals. These proportions were compared with the proportion of subjects at risk of cardiometabolic disease using the gold standard *Y*. The presence of an overlap in the 95% confidence intervals using the diagnostic test and the gold standard would suggest that the two proportions are statistically equal. The confidence interval results are displayed in Table 3.5 below.

Table 3.5: Individual ROC Cut-Point and ZBIN 95% Confidence Intervals

Combination	ROC 95% Confidence Interval	ZBIN 95% Confidence Interval
NGS	0.248 – 0.329	0.117 – 0.201
BMI	0.227 – 0.322	
AGE	0.488 – 0.574	

From this table, none of the diagnostic variables accurately identified the proportion of subject at risk, as their 95% confidence intervals did not overlap with the gold standard 95% confidence interval. However, the upper bound of the latter interval is relatively close to the lower bounds of the BMI and NGS 95% confidence intervals. This shows that ROC curves may be too simplistic in only allowing for a single diagnostic variable unadjusted for covariates to build a diagnostic tool. Chapter Four presents a method for diagnostic test construction that adjusts for covariates and uses complex survey data.

Another drawback of ROC curves is that generally they are only applicable in the case of a single diagnostic variable, which can potentially be too vague to accurately be used as a diagnostic test. It is possible that a better indicator of disease is constructed from several binary variables. The statistical method proposed in Chapter Five allows for the construction of diagnostic tests based on multiple binary diagnostic variables and it involves multiple optimal cut-points. Moreover, the method is flexible enough that it can be applied to simple random samples or more complex samples such as survey data.

Thus, although a ROC curve is one of the best-known classification methods in previous literature for diagnostic tests, it has deficiencies for survey data. ROC curves have a lack in methods for determining the optimal cut-points using the two indices in complex sample surveys, may not adjust for covariates in complex survey samples, and may not be used in determining diagnostic tests based on multiple cut-points simultaneously. The first issue was addressed in this chapter, while the remaining two issues will be addressed in chapters four and five, using other methods that allow for the use of a survey design with weights, adjust for covariates, and can be used to determine diagnostic tests based on multiple cut-points simultaneously.

CHAPTER FOUR

SINGLE CUT-POINTS FROM LOGISTIC REGRESSION FOR SURVEY DATA

4.1 Introduction

This chapter aims to explore diagnosing an individual's disease status based on the value of a single variable using an alternative approach to the ROC curve. Specifically, a logistic regression model for complex survey data will be considered, where the diagnostic variable is one of the explanatory variables. The diagnostic variable is dichotomized at a range of candidate cut-points and minimizing an information criterion such as the Akaike Information Criterion (AIC) will determine the optimal cut-point. Thus, the proposed method is applicable to complex survey data and it can adjust for covariates. However, the method determines the optimal cut-point only for one diagnostic variable at a time. This method will be extended in the next chapter, by considering several diagnostic variables simultaneously and improving the accuracy of the resulting diagnostic test.

4.2 Statistical Method

4.2.1 Logistic Regression

Classifying an individual's risk for cardiometabolic disease creates a binary response variable taking on two categories: "at risk" or "not at risk." In a statistical setting, these categorical variables must be coded numerically, generally taking values 0 and 1 for not at risk and at risk, respectively. Denote by Y this binary response variable,

classifying a subject into one of two categories: 0 or 1. The classification problem is aided by one or more explanatory variables *X*. The linear regression model

$$Y = X^T\beta + \varepsilon \tag{4.2.1}$$

where β is the linear regression coefficient, is not an appropriate model when *Y* is binary. For example, such a model can predict values of *Y* that are not 0 or 1. Therefore, a model that results in only values of 0 or 1 for *Y* is necessary. One method of statistical data analysis that implements a binary response variable is the logistic regression model.

Logistic regression uses a binary response variable and predicts the probability of the response event "1." In simple logistic regression, *X* is a single explanatory variable, and the response *Y* has a Bernoulli(p(*x*)) distribution, where p(*x*) is the probability of an individual with covariate value *X*=*x* having response 1 (e.g. Kutner et al., 2005):

$$\mathrm{P}(Y = 1) = \mathrm{p}(x) \tag{4.2.2}$$

$$\mathrm{P}(Y = 0) = 1 - \mathrm{p}(x) \tag{4.2.3}$$

One common form of the probability function is the logit function, defined as

$$\mathrm{logit}(p) = \ln \frac{p}{1-p} \tag{4.2.4}$$

For the case of the probability function, the logit scale can be written as

$$\mathrm{logit}[\mathrm{p}(x)] = \beta_0 + x\beta_1 \tag{4.2.5}$$

This can be reworked to evaluate p_i, the probability that individual *i* with covariate value x_i has response value $y_i = 1$, as such

$$p_i = \exp(\beta_0 + x_i\beta_1) \Big/ 1 + \exp(\beta_0 + x_i\beta_1) = P(\beta_0 + x_i\beta_1) \tag{4.2.6}$$

To obtain the parameter estimates β, the likelihood function is maximized, where for *n* individuals in the sample,

$$L = \prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i} \tag{4.2.7}$$

For better numerical stability, a log transformation of the likelihood function is taken and thus is maximized instead. The log-likelihood function is given as such

$$\log L = \sum_{i=1}^{n} y_i * \log P(\beta_0 + x_i\beta_1) + \tag{4.2.8}$$

$$(1 - y_i) * \log(1 - P(\beta_0 + x_i\beta_1))$$

$$= \sum_{i=1}^{n} y_i(\beta_0 + x_i\beta_1) - \sum_{i=1}^{n} \log(1 + \exp(\beta_0 + x_i\beta_1))$$

Maximizing the likelihood function (or log-likelihood function) produces estimates of the β parameters.

In multiple logistic regression, a determined number k of explanatory variables can be set (k = 1, 2, …). For k explanatory variables, there are k+1 of β parameters (β_0, β_1, …, β_k). Denote vector β and matrix X as such:

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}, \quad X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ X_{11} & X_{21} & \ldots & X_{n,1} \\ \vdots & \vdots & \vdots & \vdots \\ X_{1,k} & X_{2,k} & \cdots & X_{n,k} \end{bmatrix} \tag{4.2.9}$$

(4.2.10)

Here,

$$X^T\beta = \beta_0 + \beta_1 X_{,1} + \ldots + \beta_k X_{,k} \tag{4.2.11}$$

As before, Y_i follow independent Bernoulli(p_i) distributions, with p_i being the probability that individual i with covariate values x_i has response value $y_i = 1$ and is calculated as such

$$p_i = \exp(x_i{}^T\beta) \Big/ 1 + \exp(x_i^T\beta) \tag{4.2.12}$$

This gives logit transformation

$$\log\left(\frac{p_i}{1-p_i}\right) = x_i^T \beta \tag{4.2.13}$$

and log-likelihood function

$$\log L(\beta_0, \beta_1, \ldots, \beta_k) = \sum_{i=1}^{n} y_i (x_i^T \beta) - \sum_{i=1}^{n} \log\left(1 + \exp(x_i^T \beta)\right) \tag{4.2.14}$$

The log-likelihood is maximized numerically to estimate the parameter values β.

Logistic regression can be applied to a complex survey design (Lohr 2010). Using the notation above, if *N* is the population size then the population likelihood is defined as

$$L_U(\beta) = \prod_{i=1}^{N} p_i^{y_i} (1 - p_i)^{1-y_i} \tag{4.2.15}$$

and the log likelihood is

$$\log L_u = \sum_{i=1}^{N} y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i) \tag{4.2.16}$$

$$= \sum_{i=1}^{N} y_i (x_i^T \beta) - \sum_{i=1}^{N} \log\left(1 + \exp(x_i^T \beta)\right)$$

However, only a complex survey sample is observed with sample weights w_i. Then, the sample log-likelihood becomes

$$\log L_S = \sum_{i \epsilon S} w_i [y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i)] \tag{4.2.17}$$

$$= \sum_{i \epsilon S} w_i [y_i (x_i^T \hat{\beta}) - \log\left(1 + \exp(x_i^T \hat{\beta})\right)]$$

The finite population regression coefficients β above are the population-level solution to the following system of equations, obtained by setting to zero the partial derivatives of the above population log-likelihood with respect to the regression coefficients:

$$\sum_{i=1}^{N} x_{ij} [y_i - \frac{\exp(x_i^T \beta)}{1+\exp(x_i^T \beta)}] = 0 \qquad \text{for j} = 0, \ldots, \text{k} \tag{4.2.18}$$

If *S* is a sample, the logistic regression estimator $\hat{\beta}$ is obtained as the sample-level solution to the system of equations

$$\sum_{i \epsilon S} w_i x_{ij} [y_i - \frac{\exp(x_i^T \hat{\beta})}{1+\exp(x_i^T \hat{\beta})}] = 0 \qquad \text{for j = 0, …, k} \tag{4.2.19}$$

where w_i is the weight of the i^{th} observation in the sample, and therefore describes the number of observations in the population that unit i represents. The estimates $\hat{\beta}$ are obtained numerically, as they do not have a closed-form expression.

4.2.2 Taylor Series Linearization

Once the $\hat{\beta}$ estimated coefficients are calculated, a linearization variance estimator is found (e.g. Binder, 1983). In SAS, the procedure surveylogistic uses Taylor series linearization by default, or replication methods (if specified) for variance estimation under complex sampling designs (SAS Institute Inc. 2017). Under Taylor series linearization, simple or partial derivatives of a non-linear function are taken to linearize it locally, which can then be used to estimate the variance more easily.

For Taylor series linearization under a complex survey design, let Y be the response variable, with categories 1, 2, …, D, D + 1. Note that D=0 for the logistic regression with a binary response variable. Under stratification, let h be the stratum number for a total of H strata, i be the cluster index within stratum h (i = 1, 2, …, n_h), and j be the unit index within cluster i of stratum h (j = 1,2, …, m_{hi}). Let n be the total sample size ($n = \sum_{h=1}^{H} \sum_{i=1}^{n_h} m_{hi}$). Also let y_{hij} be the D-dimensional column vector comprised of indicator variables for the first D categories of the variable Y. The d^{th} element (d = 1, 2, …, D) is 1 if the response of the j^{th} unit of the i^{th} cluster in stratum h falls in category d, and all other remaining elements are 0.

For other notations, let w_{hij} be the sample weights and $\mathbf{D}_{hij}$ be the matrix of partial derivatives of the link function g (e.g. logistic) with respect to β. Also let π_{hij} be the expected vector of the response variable ($\pi_{hij} = E(y_{hij}|x_{hij}) = (\pi_{hij1}, \pi_{hij2}, \dots, \pi_{hijD})'$). Then $\hat{\mathbf{D}}_{hij}$ and $\hat{\pi}_{hij}$ are evaluated at $\hat{\beta}$. Denote f_h as the sampling rate for stratum h.

With these notations, the estimated covariance matrix for parameters $\hat{\beta}$ using the Taylor series method is

$$\hat{V}(\hat{\beta}) = \hat{Q}^{-1}\hat{G}\hat{Q}^{-1} \tag{4.2.20}$$

where

$$\hat{Q} = \sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\hat{D}_{hij}(diag(\hat{\pi}_{hij}) - \hat{\pi}_{hij}\hat{\pi}_{hij}')^{-1}\hat{D}_{hij}' \tag{4.2.21}$$

$$\hat{G} = \frac{n-1}{n-p}\sum_{h=1}^{H}\frac{n_h(1-f_h)}{n_h-1}\sum_{i=1}^{n_h}(e_{hi.} - \bar{e}_{h..})(e_{hi.} - \bar{e}_{h..})' \tag{4.2.22}$$

$$e_{hi.} = \sum_{j=1}^{m_{hi}} w_{hij}\hat{D}_{hij}(diag(\hat{\pi}_{hij}) - \hat{\pi}_{hij}\hat{\pi}_{hij}')^{-1}(y_{hij} - \hat{\pi}_{hij}) \tag{4.2.23}$$

$$e_{h..} = \frac{1}{n_h}\sum_{i=1}^{n_h} e_{hi}. \tag{4.2.24}$$

This Taylor series method can be used to estimate the standard errors of the $\hat{\beta}$ estimator coefficients.

4.2.3 Explanatory Variables

In logistic regression, several explanatory variables X may be available, among which one is considered a diagnostic variable. In the NHANES application, such an explanatory variable will be used to "diagnose" an individual of their cardiometabolic risk status. The binary response variable Y serves as the gold standard that classifies a subject in one of two categories (i.e. at risk or not at risk). Let X_1 be the diagnostic explanatory variable, and $X_2, \ldots, X_p$ be non-diagnostic explanatory variables. X_1 is typically quantitative, either continuous or otherwise having a large number of numerical categories, and it will be dichotomized at an optimal cut-point to classify the subjects in one of the two categories above. The diagnostic test based on such a diagnostic variable is typically easier to implement than the gold standard. It is assumed that X_1 is statistically significant in a logistic regression model of Y on the explanatory variables X_1, $X_2, \ldots, X_p$ in order to justify its use in the construction of the diagnostic test. Some of the remaining explanatory variables $X_2, \ldots, X_p$ may be statistically significant, although it is not required. This logistic regression can be applied to simple random samples of subjects, or to more complex samples such as survey data.

Once the diagnostic variable has been chosen, an optimal cut-point will be established, so that an individual's specific value can be compared to see whether they fall above or below/at that cut-point c_1. Denote the binary version of X_1 as $X_1^{c_1}$. If the sign of the estimated logistic regression coefficient for the diagnostic variable X_1 is positive, the binary diagnostic variable $X_1^{c_1}$ is defined as 1 if X_1 exceeds the cut-point c_1 and 0 otherwise. On the other hand, if the sign of the estimated logistic regression

coefficient for the diagnostic variable X_1 is negative, the binary diagnostic variable $X_1^{c_1}$ is defined as 1 if X_1 is smaller than the cut-point c_1 and 0 otherwise. Ultimately, the goal is to find the optimal cut-point c_1 so that the resulting binary diagnostic variable $X_1^{c_1}$ can be used in the construction of an accurate diagnostic test. To achieve this goal, for any candidate cut-point c_1, consider the logistic regression model of Y on $X_1^{c_1},\ldots,X_p$ and denote by $S(c_1)$ the value of an information criterion for the final logistic regression model in a model selection process that is required to retained the binary diagnostic variable but may discard some of the remaining explanatory variables X_2, …, X_p. Using a model selection process may be computationally intensive, but is necessary in order to find the best model. Finally, the optimum cut-point is selected as the candidate cut-point c_1 where S is minimized.

From a practical decision-making standpoint, it is helpful if the candidate cut-point c_1 belongs to a finite set. For example, a physician would find it more useful to use a diagnostic test where the cut-points are rounded to a small number of digits instead of having an excessive number of digits. Thus, rounding the cut-point to a small number of digits provides a practically useful finite set. More generally, the candidate cut-point c_1 is chosen in a set of equally spaced values within a large enough interval of values for the diagnostic variable X_1. Therefore, the optimum cut-point c_1 is identified where $S(c_1)$ is minimized. A direct method to identify it is to compute $S(c_1)$ for all c_1 in the finite set and choose the cut-point where S is minimized based on some chosen information criterion comparison.

One information criterion used to compare cut-points was the Akaike Information Criterion (AIC). For a given model, let k be the number of parameters, and L be the maximum value of likelihood function. Then the AIC value is given by

$$\text{AIC} = 2k - 2\ln(L) \tag{4.2.25}$$

AIC is an indicator of model quality as it penalizes a high number of parameters and rewards goodness-of-fit through the likelihood function. The "best" model is that with the smallest AIC. Thus, the cut-point that produced the smallest AIC was deemed the optimal cut-point. More details on AIC can be found in Kutner et al. (2005). While this research shows results for AIC, a number of other information criteria can also be explored, such as the Bayesian Information Criterion (BIC) (having a bigger penalty than AIC), the Deviance Information Criterion, the Hannan–Quinn Information Criterion, etc.

Once an optimal cut-point that gave the smallest AIC was identified, verification methods were used to ensure that this established cut-point was, in fact, an accurate predictor of cardiometabolic risk.

The first verification method was based on confidence intervals. In this case, the gold standard was the response binary variable Y as an indicator of an accurate test. For the binary diagnostic variable at the optimal cut-point, a 95% confidence interval for the subjects detected at risk was computed. Then it was compared against the gold standard Y's 95% confidence interval for at-risk subjects, to verify an overlap. The presence of an overlap in the intervals is consistent with the two proportions being statistically equal and verifies that the diagnostic test has comparable accuracy to that of the gold standard method.

The second verification method was analyzing the sensitivity and specificity values of the binary diagnostic variable. When logistic regression is performed using one dichotomous diagnostic variable, it is similar to finding the odds in a 2 x 2 contingency table. Comprise this 2x2 contingency table with two rows (positive or negative diagnostic test) and two columns (true positive or true negative disease status), as such

	Positive Disease Status	**Negative Disease Status**
Positive Diagnostic Test	(1,1)	(1,2)
Negative Diagnostic Test	(2,1)	(2,2)

Then sensitivity, the true positive rate, is calculated as the ratio of cell (1,1) to the total in column 1, representing those that had both a positive diagnostic test and actual positive disease status compared to all subjects with positive disease status. Similarly, specificity, the true negative rate, is calculated as the ratio of cell (2,2) to the total in column 2, representing those that had both a negative diagnostic test and actual negative disease status compared to all subjects with negative disease status. An optimal binary diagnostic variable would have high sensitivity and specificity values. These ratios are estimated by incorporating both the survey design and weights, as discussed in the previous chapter.

In summary, the chosen diagnostic variable would potentially be an indicating marker of the response variable Y (in this case, cardiometabolic risk status). After being dichotomized around potential cut-points, an optimal cut-point c_i was selected as that giving the smallest AIC (i.e. minimized $S(c_i)$). Then, the optimal cut-point underwent two verification methods: (1) searching for the presence of overlapping confidence

intervals with that of the gold standard and (2) having high sensitivity and specificity values, to verify that it was accurate in diagnosing the response status.

4.3. 1-Dimensional Simulation Study

To verify that the proposed method correctly identifies the optimal cut-points for a single diagnostic variable under various survey designs, a simulation study was performed. In this simulation study, artificial data from prespecified methods was created and compared to the true values to verify that logistic regression is an accurate method in choosing an optimal cut-point. The three survey designs considered were: simple random sampling, stratified sampling, and two-stage cluster sampling. These survey designs are building blocks for more complex survey design, such as NHANES. All computations in this thesis were performed on a Windows 11 system with 8GB RAM and 1.80GHz i5 processor. A simulation was run with a sample size of 1000, comparable to the sample size of the NHANES data set, to determine the optimal cut-point under each survey design. For each of these different sampling designs, the sample was selected from a population of size 100,000. In addition, the simulation study included five explanatory variables.

In the first design, each subject had a weight equal to 100 and the sample was selected using a simple random sample. The second design was stratified sampling, where the population was divided into four strata of equal size, with 250 subjects randomly selected from each stratum. This resulted in a self-weighting sample, each subject in the sample having a weight equal to 100. The following values were added to

the five explanatory variables: 2 for stratum 1, 1.5 for stratum 2, 1 for stratum 3, 0.5 for stratum 4 to induce clearly-defined distinctions between each stratum. These numbers were not kept too large, as larger values could create too strong of a difference, affecting the parameter estimates. The third design was a two-stage cluster sampling, where the population consists of 80 PSUs of size 1250 each. Within each PSU, a compound symmetry correlation of parameter 0.2 was induced among the values of the explanatory variables to mimic correlation that exists within cluster groups. This correlation parameter was kept fairly low, as too strong of a correlation (nearing 1) may not be realistic. A sample of 8 PSUs were selected randomly, and within each sampled PSU, 125 SSUs were sampled randomly giving an overall two-stage cluster sample of size 1000. This sample was also self-weighted, with 100 the weight of each unit in the sample.

For all three sampling designs, the true regression coefficients were 4, -2, 3, 0.1, -0.01 and the intercept was -5. These were chosen to be comparable in number to a N(0,1) distribution, which ranges mostly from -1.96 to 1.96. The values of 4 and -5 indicate more strongly significant explanatory variables. The simulation tested a single diagnostic explanatory variable, with the true cut-point set to its 25^{th} sample quantile. The resulting binary diagnostic variable together with the last four explanatory variables above were used to compute the probabilities of success for each subject in the sample using the logistic link function. Then the gold standard response variable was obtained by simulating from the Bernoulli distribution with the resulting probabilities of success for each subject. This simulation process was repeated 100 times for the response variable to identify the optimal cut-points. For each sampling design, the interval for the candidate

cut-points was defined by the 0.01 and 0.99 sample quantiles of the diagnostic variable, then discretized in subintervals of length 0.01. The optimal cut-point was searched in the resulting finite set of cut-points. This was achieved by fitting logistic regression models to each simulated binary response variable, including the diagnostic variable dichotomized at each candidate cut-point, while also performing a model selection on the remaining four explanatory variables.

The function ‘svyglm’ in the ‘survey’ package (Lumley, 2020) has been used to fit logistic regression models for survey data. To determine the optimal cut-point for the single diagnostic variable, the AIC value was computed on the finite set at each potential cut-point value, and the optimal cut-point was deemed as that which gave the smallest AIC value under each survey design. The function ‘response.AIC’ reports the cut-point that resulted in the minimum AIC value for each simulation. Rarely, this function can achieve its minimum at more than one candidate cut-point, since the minimum AIC could occur at multiple cut-point values. To decide between these values, the minimum of these cut-points (the first candidate cut-point resulting in the minimum AIC value) was used. Figure 4.3.1 displays the range of AIC values to find which cut-point gave the smallest AIC under each survey design, where the x-axis represents the candidate cut-points for the diagnostic variable and the y-axis represents their corresponding AIC values. The cut-points are not viewed as additional parameters in the likelihood function, but they are obtained as a result of a model selection process through the minimization of an information criterion such as AIC.

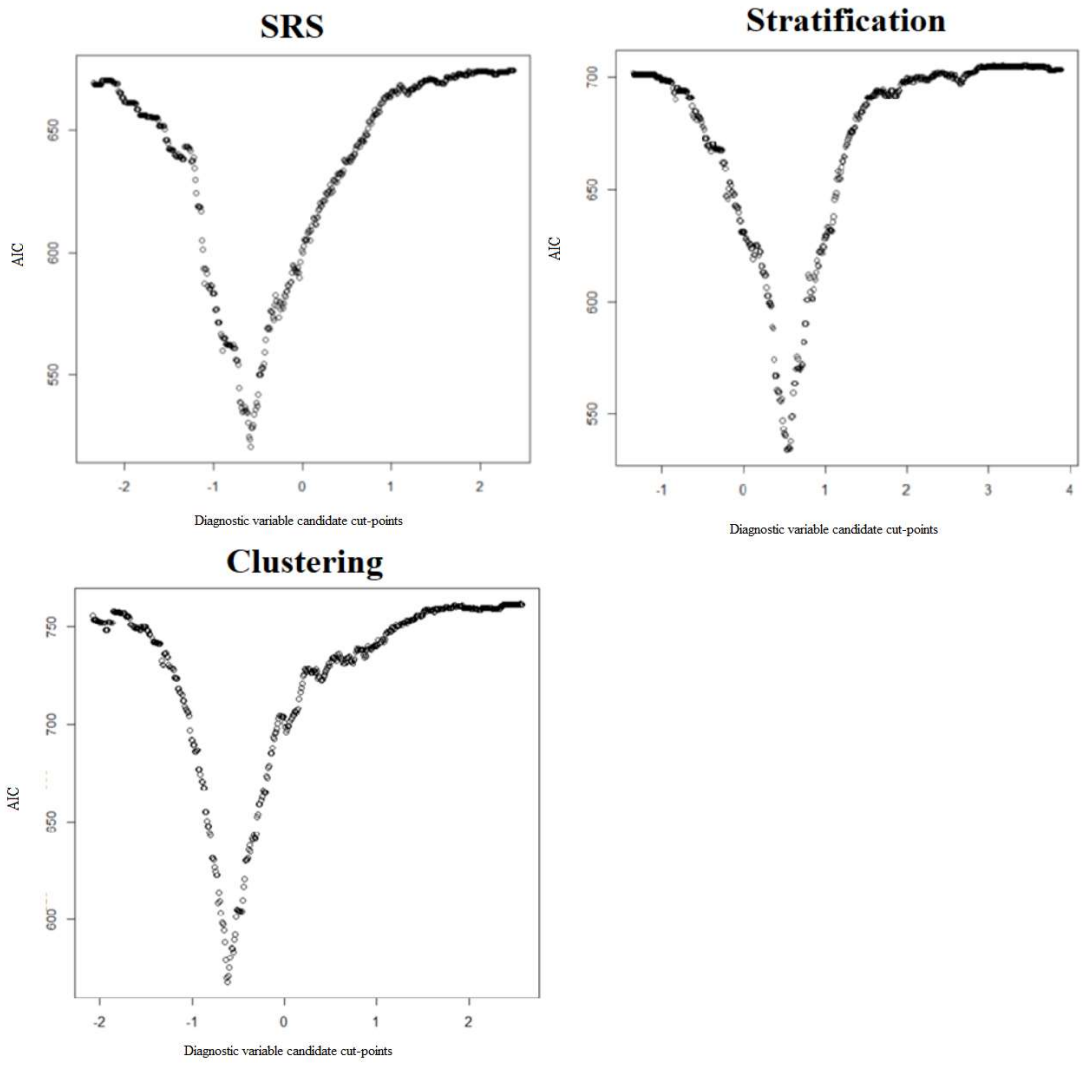

Figure 4.3.1: 1-Dimensional Simulation AIC Plots

Table 4.1: 1-Dimensional True 25% Cut-Points

Survey Design	SRS	Stratification	Clustering
True cut-point	-0.58	0.56	-0.61

From these figures, the simulated optimal cut-points at the smallest AIC value were close to the true 25th quantile cut-points for each survey design, given in Table 4.1. The 25th quantiles were negative for SRS and clustering because the averages of the diagnostic

variables were close to zero, while for stratification is shifted more to the positive side of zero, since positive means were added in each stratum.

Histograms displaying the simulation distribution of the identified optimal cut-points for each of the survey designs are displayed in Figure 4.3.2.

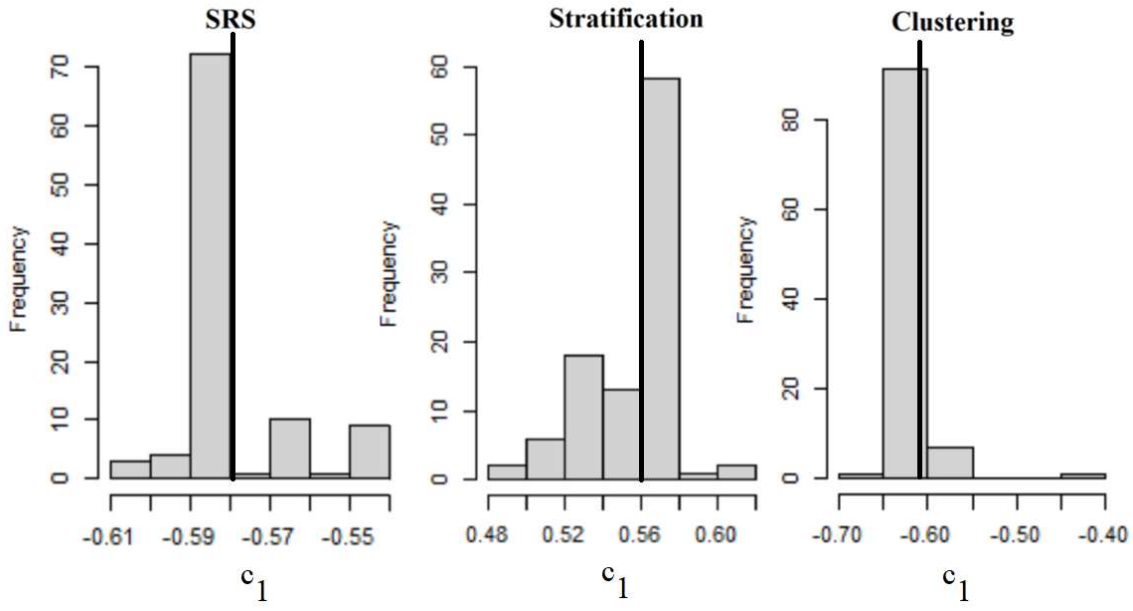

Figure 4.3.2: 1-Dimensional Simulation Cut-Point Histograms

For all three sampling designs, the centers of the histograms are close to the true cut-points while the variability is relatively small. Thus, the logistic regression model accurately identified the true cut-points under all three survey designs.

Summaries of the cut-points simulation distributions obtained through each of the three sampling designs over all 100 simulations were calculated and compared to the true cut-points set at the 25^{th} sample quantile. These results are displayed in Table 4.2.

Table 4.2: Summary of 1-Dimensional Optimal Cut-Points Simulation Results

Survey Design	SRS	Stratification	Clustering
Sample Mean	-0.58	0.56	-0.61
Average Absolute Deviation	0.01	0.02	0.01
Sample Standard Deviation	0.01	0.02	0.02
Sample Median	-0.58	0.57	-0.61
Median Absolute Deviation	0.00	0.00	0.00

The total computation time for all three survey designs was approximately 11.2 hours. Thus, the direct search method can be quite computationally intensive since all cut-points were tested on a relatively fine 1-dimensional grid. However, for all three survey designs, the sample mean and sample median was close to the true cut-point, while accounting for their uncertainty. The variability measures are relatively small relative to the cut-points values, supporting further the accuracy of the method. The median absolute deviation is unrealistically small because many of the identified cut-point values are equal among themselves, as the histograms in Figure 4.3.2 indicate. Overall, the simulation study provides empirical evidence that, although computationally expensive, the direct method identifies accurately the optimal cut-points.

4.4 1-Dimensional Cardiometabolic Risk Application

The NHANES data set was used to fit a logistic regression for survey data to the binary CMR score and several explanatory variables, as shown in Table 4.3 below.

Table 4.3: Results from the logistic regression model of the response binary CMR variable on covariates

Parameter	Estimate	SE	t-value	P-value
Intercept	-11.8356	1.7879	-6.6200	<0.0001**
NGS	-2.8065	1.2956	-2.1662	0.0432*
Survey cycle	0.2762	0.1567	1.7624	0.0941
Sex	0.8300	0.1714	4.8424	0.0001**
Age	0.1402	0.0539	2.6011	0.0175*
BMI	0.2958	0.0464	6.3760	<0.0001**
Race/ethnicity (ref=other race)				
Mexican American	0.6097	0.2414	2.5257	0.0206*
Other Hispanic	0.6680	0.3522	1.8969	0.0732
White non-Hispanic	0.7145	0.3250	2.1984	0.0405*
Black non-Hispanic	-0.6150	0.3817	-1.6110	0.1237
Asian non-Hispanic	0.9340	0.4529	2.0624	0.0531
MNH	-0.0230	0.0347	-0.6621	0.5159
PIR	0.0387	0.1099	0.3524	0.7284
TSA	0.0004	0.0008	0.5137	0.6134

Note: * $p < 0.05$, ** $p < 0.001$. NGS = normalized grip strength, BMI = body mass index, MNH = meals not eaten at home, PIR = poverty to income ratio, TSA = time in sedentary activity

The following variables were found to be statistically significant and were individually considered as the diagnostic variables: (1) normalized handgrip strength (NGS), (2) body mass index (BMI), and (3) age. To find the optimal cut-point for each of them, the variables at hand were (one at a time) dichotomized at candidate cut-points, and logistic regression models were fitted on the single dichotomized variable along with the remaining explanatory variables. NGS was tested over the range{0.30, 0.31, …, 0.45}, BMI over {17, 18, …, 37} , and AGE over {12, 13, …, 24}. The iterative searches were run individually on each set of candidate cut-points to select the corresponding cut-point that produced the smallest AIC. For example, NGS was dichotomized over values of 0.30, 0.31, …., 0.45, and then the AIC value at each candidate cut-point was recorded.

Then the overall best cut-point was deemed as that which resulted in the smallest AIC. This was then repeated for diagnostic variables BMI and AGE.

Figure 4.4.1 displays the AIC values for each of the potential cut-point values of each diagnostic variable under logistic regression. Thus, each iterative search resulted in individual optimal diagnostic cut-point values of $c_1 = 0.36$ for NGS, $c_2 = 27$ for BMI, and $c_3 = 21$ for AGE, as they each resulted in the lowest AIC value of 532.0855, 548.2159, and 527.4602, respectively. Note that the cut-point of 18 for AGE had an AIC only marginally larger than 21 for AGE, at 527.5586. From this analysis, only the BMI individual cut-point matched that of the ROC curves from Chapter Three, whereas NGS and AGE resulted in different optimal cut-points. Thus, ROC curves may not always match logistic regression results.

The method to find the cut-point for NGS somewhat aligns with the method of a previous study by Brown et al. (2020), which also used 2011-2012 and 2013-2014 NHANES data to determine the 1-dimensional logistic regression cut-point of NGS. However, the NGS cut-point was used a screening test for Type 2 diabetes, as compared to any cardiometabolic risk. In addition, their study focused on adults aged 20-80 years (as compared to younger subjects aged 12-24 years) without common diabetes comorbidities. The optimal NGS cut-points were as such: 0.78 for young male participants (20-50 years old), 0.57 for young female participants, 0.68 for older male participants (50-80 years old), and 0.49 for older female participants. Thus, males tend to be stronger than females at all adult ages.

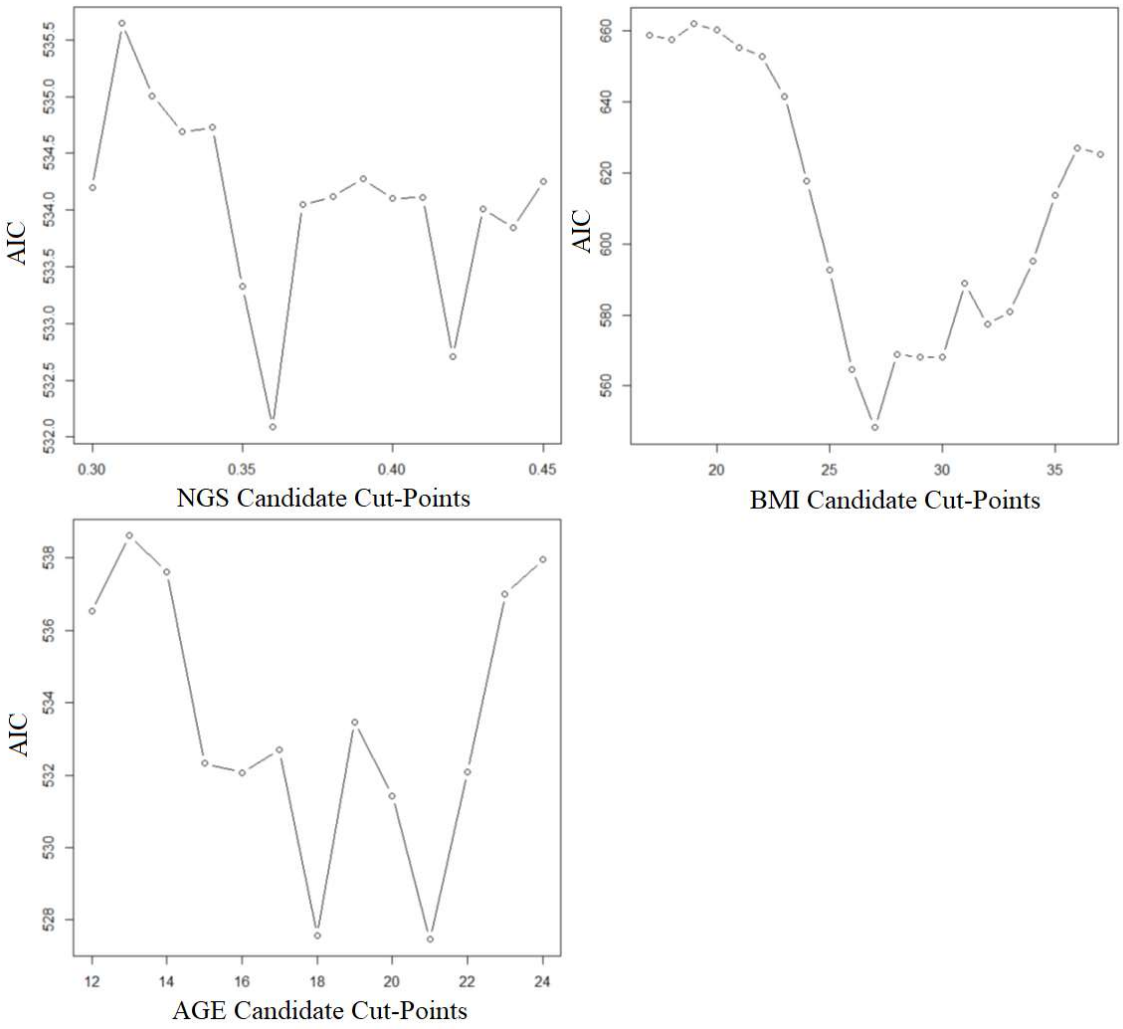

Figure 4.4.1: Logistic regression candidate cut-points versus AIC

In another study by DeHondt et al. (2023), the 1-dimensional logistic regression NGS cut-point was also determined from 2011-2012 and 2013-2014 data. This study did focus on younger respondents aged 12-24 years (n=1,033), and even further separated them into males and females and two age categories. With this study, the NGS cut-points for participants aged 12-17 years was 0.39 for males and 0.34 for females, and the NGS cut-points for participants aged 18-24 years was 0.45 for males and 0.34 for females.

Thus, males tend to be stronger than females at adolescent ages. These cut-points are in alignment with the total 1-dimensional cut-point of 0.36.

Next, the whole-sample proportions of subjects at high cardiometabolic risk using both the dichotomized diagnostic variables at optimal cut-points and the gold standard Y were investigated to determine if there was any overlap in their 95% confidence intervals, suggesting that the two proportions would be statistically equal. Furthermore, the whole-sample sensitivity and specificity were calculated for the individual binary diagnostic tests and binary cardiometabolic risk scores. Analyses were performed both in R and SAS. For simplicity, only the R results will be displayed below. Although the SAS results were similar, it is important to note that the default sensitivity and specificity values were swapped between SAS and R, likely due to R and SAS defining successes and failures differently. Tables 4.4 and 4.5 display the 95% confidence intervals and sensitivity and specificity levels for each of the individual binary diagnostic tests.

Table 4.4: Individual Diagnostic Test and ZBIN 95% Confidence Intervals

Combination	**Test 95% Confidence Interval**	**ZBIN 95% Confidence Interval**
NGS	0.076 – 0.118	0.117 – 0.201
BMI	0.227 – 0.322	
AGE	0.263 – 0.342	

Table 4.5: Individual Diagnostic Test Sensitivity and Specificity Levels

Combination	**Sensitivity**		**Specificity**	
	Estimate	**Std. Error**	**Estimate**	**Std. Error**
NGS	0.319	0.033	0.945	0.009
BMI	0.852	0.032	0.835	0.019
AGE	0.451	0.053	0.726	0.022

The 95% confidence interval for the proportion of subjects at risk using the gold standard Y was 0.117 - 0.201. Among the three diagnostic tests that consist of individual binary diagnostic variables, BMI had the best combination of sensitivity and specificity, both above 0.8, but its 95% confidence interval 0.227 – 0.322 for the proportion of subjects classified at risk did not overlap with the above 95% confidence interval for the proportion of subjects at risk using the gold standard Y, suggesting that two proportions may not be statistically equal. The 95% confidence interval for NGS overlapped with the gold standard interval, but its sensitivity was very low at 0.319. This indicates that using NGS alone may not be the best diagnostic test to determine whether an individual is truly at risk of cardiometabolic disease. Using AGE alone was the worst diagnostic test, since its 95% confidence interval did not overlap with the corresponding gold standard confidence interval, and both its sensitivity and specificity values were low. Thus, none of the diagnostic tests based on individual binary diagnostic tests were accurate enough, indicating that a more complex diagnostic test may be necessary to provide an accurate predictor of cardiometabolic risk status. Such a diagnostic test will be explored in the next chapter, constructed from several diagnostic variables simultaneously instead of separate single diagnostic variables.

CHAPTER FIVE

MULTIPLE CUT-POINTS FROM LOGISTIC REGRESSION FOR SURVEY DATA

5.1 Introduction

It is beneficial to determine cardiometabolic risk status based off a single cut-point diagnostic variable, due to its simplicity to administer and implement. However, it is much more comprehensive to investigate simultaneously several diagnostic variables that may contribute to cardiometabolic risk in order to determine a more accurate diagnostic test. Indicators of disease using multiple variables exist in other contexts. For example, according to the American Diabetes Association (2012), diagnostic criteria classify a patient as having type 2 diabetes if fasting plasma glucose exceeds 126 mg/dL, or 2-hour plasma glucose exceeds 200 mg/dL during oral glucose tolerance test (OGTT), or HbA1c exceeds 6.5%.

The proposed method uses a logistic regression model involving a gold standard binary response variable and several explanatory variables, among which a group of them are designated as diagnostic variables. These explanatory diagnostic variables are assumed to be quantitative, and statistically significant. The optimum cut-points of the explanatory diagnostic variables are identified and a diagnostic test using the resulting binary diagnostic variables is constructed. To improve the computational efficiency of the method, kriging-based optimization is used to identify the optimal cut-points. Simulations and a cardiometabolic risk application using NHANES data will illustrate the method. Kriging-based optimization has been used mostly in engineering applications (e.g. Jones et al., 1998), but it is rather uncommon in biomedical applications.

In the construction of diagnostic tests based on multiple binary diagnostic variables, similar verification methods are necessary to ensure diagnostic accuracy. Thus, both overlap of confidence intervals for proportions of subjects at risk, as well as high sensitivity and specificity values will be sought out from each of the tested combinations of the diagnostic variables to determine which provides the most accurate and best diagnostic test option. When several diagnostic tests are available, it is desirable to use the test that leads to the fewest misclassifications.

5.2 Determining the Optimal Cut-Point

5.2.1 Logistic Regression Models

Consider a binary response variable Y as the gold standard that correctly classifies a subject in one of two categories, e.g. at risk of disease or not. Let $X_1, X_2, \ldots, X_k, X_{k+1}, \ldots, X_p$ be explanatory variables, where the first k are designated as diagnostic variables that will be used to construct a multi-variable diagnostic test. This diagnostic test attempts to classify the subjects in one of the two categories above, and is typically easier to implement than the gold standard. It is assumed that the diagnostic variables are statistically significant in a logistic regression model of Y on the explanatory variables $X_1, X_2, \ldots, X_k, X_{k+1}, \ldots, X_p$ in order to justify their use in the construction of the diagnostic test. In addition, it is assumed that all diagnostic variables are quantitative. Some of the remaining explanatory variables $X_{k+1}, \ldots, X_p$ may also be statistically significant, although it is not required. The logistic regression can be applied to simple random samples of subjects, or to more complex samples such as survey data.

To facilitate the classification decision-making process, the diagnostic test will be based on binary versions of the diagnostic explanatory variables, denoted by $X_1^{c_1}$,…, $X_k^{c_k}$. The value of $X_i^{c_i}$, being 0 or 1, is defined in the same manner as in the 1-dimensional case explained in Chapter Four. However, the new goal is to find a series of optimal cut-points c_i for each diagnostic variable X_i, so that the resulting binary diagnostic variables $X_1^{c_1}$,…, $X_k^{c_k}$ can be used in the construction of an accurate diagnostic test. For any vector of candidate cut-points $(c_1, \dots, c_k)$, consider the logistic regression model of Y on $X_1^{c_1}$,…, $X_k^{c_k}, X_{k+1}, \dots, X_p$ and denote by $S(c_1, \dots, c_k)$ the value of an information criterion, such as AIC, for the final logistic regression model in a model selection process that is required to retained the binary diagnostic variables but it may discard some of the remaining explanatory variables X_{k+1}, …, X_p. As before, select the optimum cut-points as the candidate cut-points $(c_1, \dots, c_k)$ where S is minimized.

Following the 1-dimensional case, it is also assumed that the candidate cut-points in the k-dimensional case belong to a finite but possibly large set. For each i in 1,…,k, choose the candidate cut-point c_i in a set of equally spaced values within a large enough interval of values for the diagnostic variable X_i. Therefore, the vectors of candidate cut-points $(c_1, \dots, c_k)$ are finite by construction and form a grid of points. The optimum cut-points $(c_1, \dots, c_k)$ are identified where $S(c_1, \dots, c_k)$ is minimized. While they are not guaranteed to be unique, it is unlikely in practice that the AIC has two or more global minima (if it does, additional criteria can be considered). A direct method to identify them is to compute $S(c_1, \dots, c_k)$ for all $(c_1, \dots, c_k)$ on the grid and choose the vector where S is minimized. However, when the grid is fine enough and/or the number k of

diagnostic variables is relatively large, fitting so many logistic regression models becomes computationally prohibitive even for data sets of moderate size. To address this computational drawback, kriging-based optimization is used, as described next.

5.2.2 Kriging-Based Optimization

Using terminology from the design and analysis of computer experiments (e.g. Sacks et al., 1989 and Santner et al., 2004), S is viewed as a computer model with input $(c_1, \dots, c_k)$ and output $S(c_1, \dots, c_k)$. The above grid of points $(c_1, \dots, c_k)$ represents the input space. Let $\left(c_1^{(1)}, \dots, c_k^{(1)}\right), \dots, \left(c_1^{(D)}, \dots, c_k^{(D)}\right)$ be a sample of inputs, chosen using a space filling design such as a maximin Latin hypercube design (e.g. Morris & Mitchell, 1995). The output values $S\left(c_1^{(1)}, \dots, c_k^{(1)}\right), \dots, S\left(c_1^{(D)}, \dots, c_k^{(D)}\right)$ are assumed to have a multivariate normal distribution of constant mean vector $\gamma \mathbf{1}_{\mathrm{D}}$ and covariance matrix $\tau^2 \mathbf{R}_{\mathrm{D}}$. This assumption can be verified, e.g. using cross-validation. The correlation between two output values $S\left(c_1^{(i)}, \dots, c_k^{(i)}\right)$ and $S\left(c_1^{(j)}, \dots, c_k^{(j)}\right)$ depends on the distance between the corresponding inputs $\left(c_1^{(i)}, \dots, c_k^{(i)}\right)$ and $\left(c_1^{(j)}, \dots, c_k^{(j)}\right)$. Here, the exponential correlation is chosen since the output may not be a smooth, differentiable function of its input vector. For example, changing the cut-points from current values to neighboring values on the grid may lead to dropping/adding explanatory variables at the model selection stage. This will change the number of parameters in the penalty term of the information criterion, inducing roughness (deviations of a surface from its ideal form) consistent with the output generated by the exponential correlation. Evidence for such

roughness is visible in Figure 4.4.1 where even the 1-dimensional plots are jagged, non-smooth.

Let $(c_1, \dots, c_k)$ be a new input. The kriging predictor of $S(c_1, \dots, c_k)$ is

$$\hat{S}(c_1, \dots, c_k) = \hat{\gamma} + \mathbf{R}_0' \mathbf{R}_D^{-1} (\mathbf{S} - \hat{\gamma} \mathbf{1}_D) \tag{5.2.1}$$

and the kriging prediction variance is

$$V(c_1, \dots, c_k) = \hat{\tau}^2 (1 - \mathbf{R}_0' \mathbf{R}_D^{-1} \mathbf{R}_0) + \hat{\tau}^2 \frac{(1 - \mathbf{R}_0' \mathbf{R}_D^{-1} \mathbf{R}_0)^2}{\mathbf{1}_D' \mathbf{R}_D^{-1} \mathbf{1}_D} \tag{5.2.2}$$

where $\hat{\gamma}, \hat{\tau}^2$ are the maximum likelihood estimators of γ, τ^2, respectively, the sampled outputs are gathered in the vector $\mathbf{S} = \left[S\left(c_1^{(1)}, \dots, c_k^{(1)}\right), \dots, S\left(c_1^{(D)}, \dots, c_k^{(D)}\right)\right]'$, while $\mathbf{R}_0$ is the vector of model correlations between $S(c_1, \dots, c_k)$ and $\left\{S\left(c_1^{(1)}, \dots, c_k^{(1)}\right), \dots, S\left(c_1^{(D)}, \dots, c_k^{(D)}\right)\right\}$. In the design and analysis of computer experiments terminology, $\hat{S}(c_1, \dots, c_k)$ is called an emulator (or metamodel) of $S(c_1, \dots, c_k)$. This emulator is a very fast approximation of the computationally intensive information criterion S because both $\hat{S}$ and its variance V involve very simple algebra. In addition, the value of D is typically small since only a limited number of information criterion evaluations, from the corresponding fitted logistic regression models, is computationally affordable.

The goal is to choose the optimal cut-points $\left(c_{1,o}, \dots, c_{k,o}\right)$ where $S(c_1, \dots, c_k)$ is minimized on the grid. A simplistic approach would be to minimize only $\hat{S}(c_1, \dots, c_k)$. However, this approach does not take into consideration the emulator variance V, which may lead to inaccurate solutions. As a more accurate alternative, the expected improvement (EI) function is maximized

$$EI(c_1, \dots, c_k) = \left(S_{min} - \hat{S}(c_1, \dots, c_k)\right) \Phi\left(\frac{S_{min} - \hat{S}(c_1, \dots, c_k)}{\sqrt{V(c_1, \dots, c_k)}}\right) \tag{5.2.3}$$

$$+\sqrt{V(c_1, \dots, c_k)}\phi\left(\frac{S_{min} - \hat{S}(c_1, \dots, c_k)}{\sqrt{V(c_1, \dots, c_k)}}\right)$$

where $S_{min} = \min\left\{S\left(c_1^{(1)}, \dots, c_k^{(1)}\right), \dots, S\left(c_1^{(D)}, \dots, c_k^{(D)}\right)\right\}$, Φ is the cumulative distribution function of $N(0,1)$ and ϕ is the probability density function of $N(0,1)$. The relationship between the two optimization problems is described in Jones et al. (1998), generally for any computer model.

5.2.3 EGO Algorithm

The EI expression above suggests that the kriging-based optimization method samples a new input where the emulator improves over previous S values (exploitation) or where its variance is large (exploration), leading to the following efficient global optimization (EGO) algorithm:

- Sample D inputs $\left(c_1^{(1)}, \dots, c_k^{(1)}\right), \dots, \left(c_1^{(D)}, \dots, c_k^{(D)}\right)$ on the grid and compute the information criterion values $S\left(c_1^{(1)}, \dots, c_k^{(1)}\right), \dots, S\left(c_1^{(D)}, \dots, c_k^{(D)}\right)$.
- Compute the fast kriging emulator $\hat{S}(c_1, \dots, c_k)$ and its variance $V(c_1, \dots, c_k)$ at any $(c_1, \dots, c_k)$ on the grid.
- Choose $\left(c_1^{(D+1)}, \dots, c_k^{(D+1)}\right)$ on the grid that maximizes the expected improvement $EI(c_1, \dots, c_k)$.

- Augment the set of sampled inputs as $\left(c_1^{(1)}, \dots, c_k^{(1)}\right), \dots, \left(c_1^{(D+1)}, \dots, c_k^{(D+1)}\right)$ and the corresponding set of outputs as $S\left(c_1^{(1)}, \dots, c_k^{(1)}\right), \dots, S\left(c_1^{(D+1)}, \dots, c_k^{(D+1)}\right)$.
- Iterate the steps above until $S\left(c_1^{(D+1)}, \dots, c_k^{(D+1)}\right) \leq S_{min}$ and the relative error $\left|\frac{S\left(c_1^{(D+1)}, \dots, c_k^{(D+1)}\right) - S_{min}}{S_{min}}\right|$ is smaller than a tolerance error ε (e.g. 0.0001). Denote by $\left(c_{1,o}, \dots, c_{k,o}\right)$ the optimal cut-points obtained at the last iteration.

Since the EGO algorithm explores a possibly large but finite grid, it will always find the optimum on the grid in a finite number of steps. The advantage of the EGO algorithm, however, is that its guided search can identify the optimum on the grid in a smaller number of steps, before the entire finite grid is exhausted, thus improving the computational efficiency. Overall, the kriging method is a faster and less computationally expensive method of searching for multiple optimal cut-points than the direct method, and the EGO algorithm uses kriging to speed up optimization. The simulations and the real application discussed in the next sections will provide evidence of such an advantage.

5.2.4 Diagnostic Test

Once the optimum cut-points $\left(c_{1,o}, \dots, c_{k,o}\right)$ have been identified on the grid, the corresponding binary variables $X_1^{c_{1,o}}, \dots, X_k^{c_{k,o}}$ will be used to construct a binary diagnostic test *T* that indicates accurately the subject's status given by *Y*. However, there may be many such possible binary diagnostic tests. For example, one possibility is to

define the diagnostic test as T=1 if $X_1^{c_{1,o}} = 1$ *and* $X_2^{c_{2,o}} = 1$ *and* … *and* $X_k^{c_{k,o}} = 1$, while T=0 otherwise. Another possibility may be to define the diagnostic test as T=1 if $X_1^{c_{1,o}} = 1$ *or* $X_2^{c_{2,o}} = 1$ *or* … *or* $X_k^{c_{k,o}} = 1$, while T=0 otherwise. Yet another possibility is to define T=1 if $X_1^{c_{1,o}} = 1$ *and* ($X_2^{c_{2,o}} = 1$ *or* … *or* $X_k^{c_{k,o}} = 1$), while T=0 otherwise. In the latter case, some of the diagnostic variable must contribute to the diagnostic test (corresponding to '*and*'), while the contribution of other diagnostic variables is not as critical (corresponding to '*or*'). An early indicator of each diagnostic variable's contribution to the diagnostic test could be the P-values from the logistic regression model of the binary response variable Y on the original explanatory variables $X_1, X_2, \ldots, X_k, \ldots, X_p$. The smaller the P-value of X_i's coefficient, the stronger the contribution of its binary version $X_i^{c_{i,o}}$ is expected to be to the diagnostic test.

Ideally, a diagnostic test perfectly indicates the status, that is T=1 if and only if Y=1. However, some classification errors are likely to occur. Three validation measures to identify an accurate diagnostic test will be used. Specifically, the diagnostic test T leading to the largest sensitivity and specificity values will be chosen. Another criterion used is to check if the proportions of subjects at risk are statistically equal, using both the diagnostic test T and the gold standard Y.

5.3 Simulation Study

To verify that kriging-based optimization accurately identifies the optimal cut-points, a simulation study was performed with a sample size of 1000, comparable to the sample size of the NHANES data set. Five explanatory variables were considered, with

values generated from the standard normal distribution, resulting in a regression matrix with five columns and 1000 rows. The three survey designs from the previous chapter were investigated: simple random sampling, stratified sampling, and two-stage cluster sampling. For each of these different survey designs, the sample was selected from a population of size 100,000.

For all three sampling designs, the true regression coefficients were set again at 4, -2, 3, 0.1, -0.01 and the intercept was -5. The explanatory variables together with the regression coefficients were used to compute the probabilities of success for each subject in the sample using the logistic link function. Then the gold standard response variable was obtained by simulating from the Bernoulli distribution with the resulting probabilities of success for each subject. This simulation process was repeated 100 times for the response variable. The simulated data under the general logistic regression model for survey data re-estimated the true regression coefficients (-5, 4, -2, 3, 0.1, -0.01). Histograms displaying the distribution of the estimated true regression coefficients for each of the survey designs are displayed in Figures 5.3.1, 5.3.2, and 5.3.3. From these figures, for all three sampling designs, the centers of the histograms closely match the true regression coefficients while the variability is not excessively large. Thus, the general logistic regression model accurately estimated the true regression coefficients under all three survey designs. The function ‘svyglm’ in the ‘survey’ package (Lumley, 2020) has been used to fit logistic regression models for survey data.

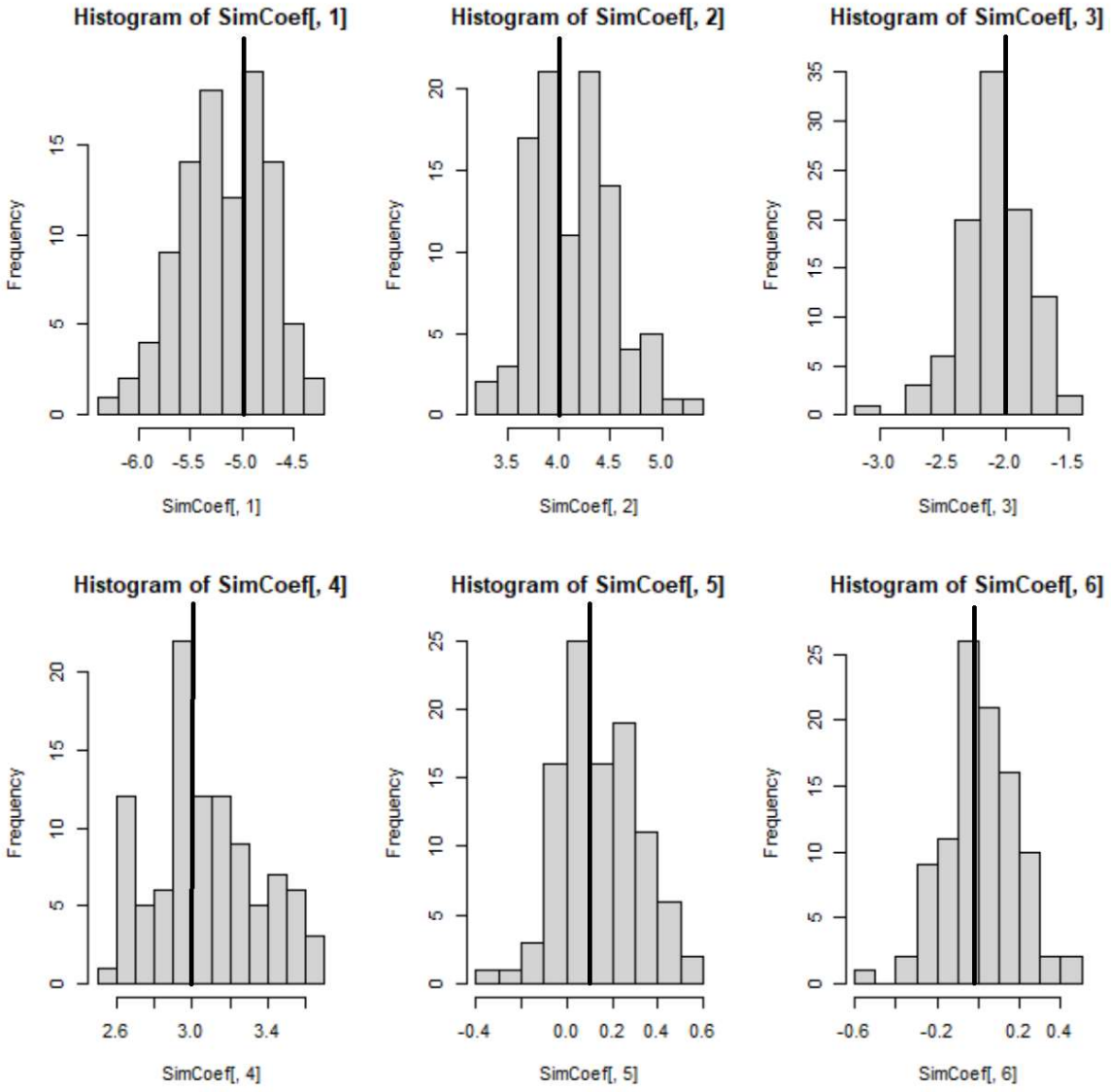

Figure 5.3.1: SRS Simulation Regression Coefficients Estimates

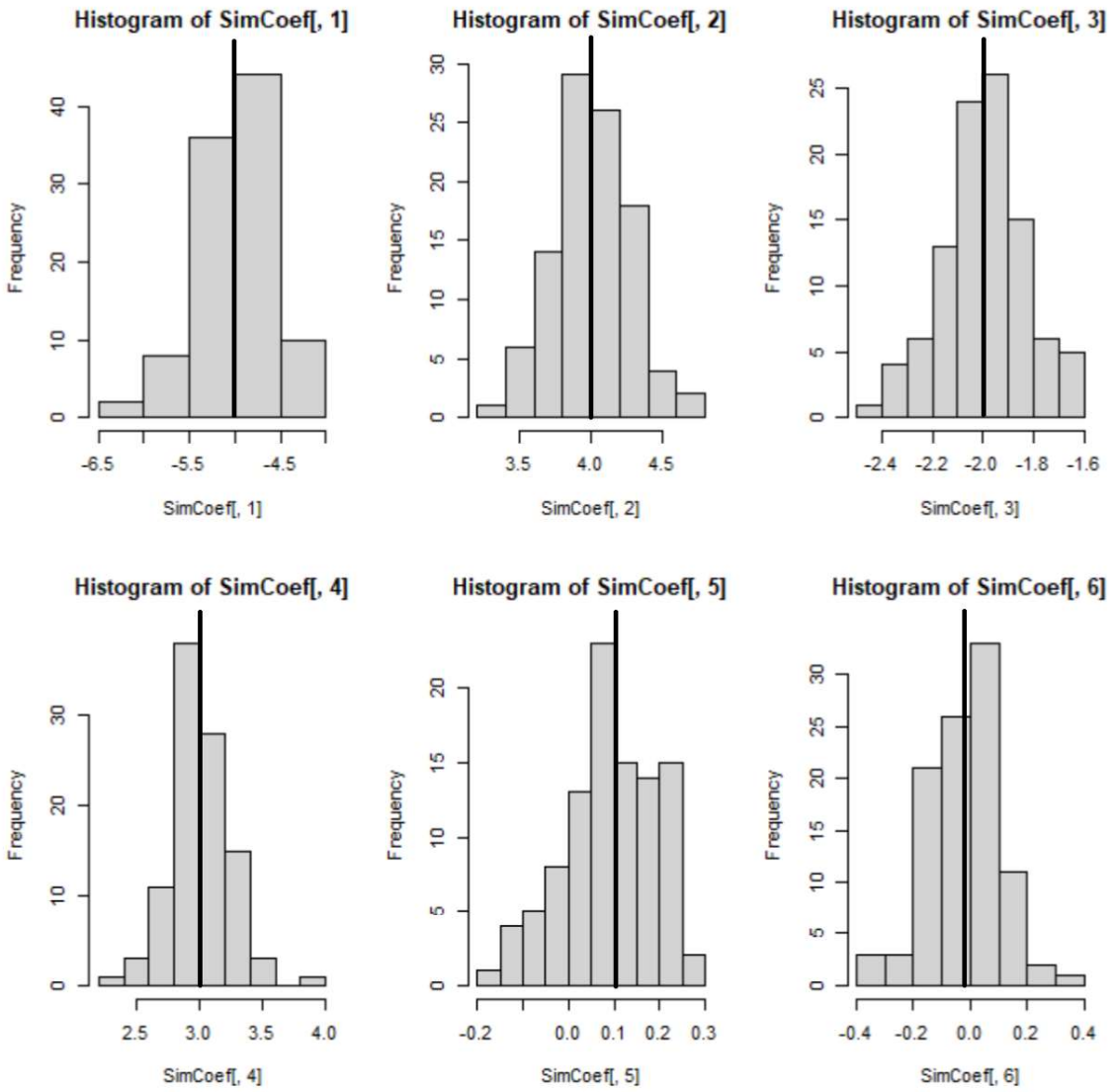

Figure 5.3.2.: Stratification Simulation Regression Coefficients Estimates

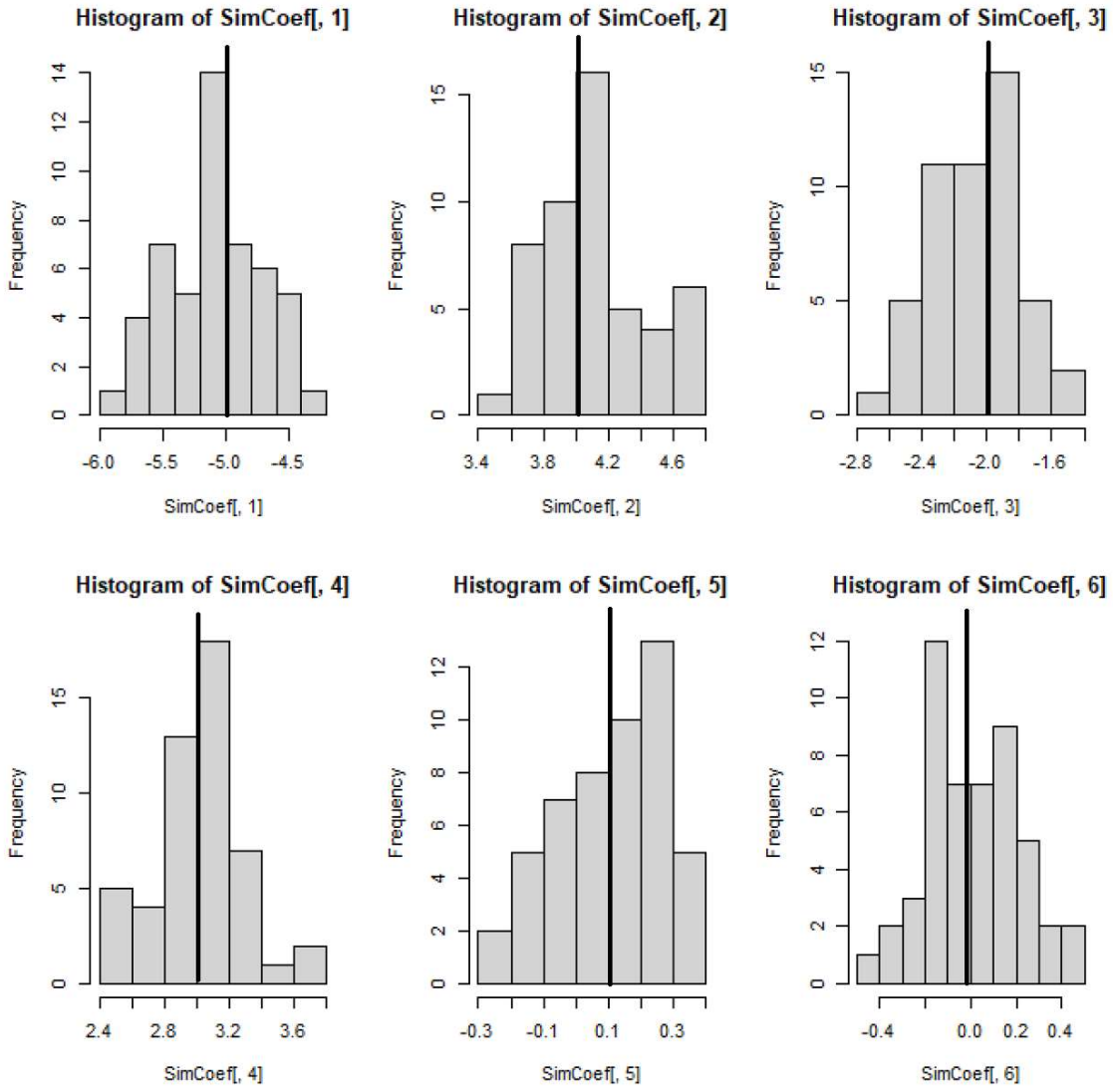

Figure 5.3.3: Clustering Simulation Regression Coefficients Estimates

Next, the first three explanatory variables were designated as diagnostic variables, and the true cut-points are set to their 0.25, 0.5 and 0.75 sample quantiles, displayed in Table 5.1.

Table 5.1: True Cut-Points

Survey Design	**SRS**			**Stratification**			**Clustering**		
	c_1	c_2	c_3	c_1	c_2	c_3	c_1	c_2	c_3
True cut-points	-0.72	0.08	0.67	0.39	1.26	2.01	-0.74	0.25	0.56

The resulting three binary diagnostic variables together with the last two explanatory variables above were used to compute the probabilities of success for each subject in the sample using the logistic link function. Similarly, the gold standard response variable was obtained by simulating from the Bernoulli distribution with the resulting probabilities of success for each subject. As above, the simulation process was repeated 100 times for the response variable, and for each simulation the kriging-based optimization method was used to identify the optimal cut-points. For each sampling design, the input space was defined by the 0.01 and 0.99 sample quantiles of each diagnostic variable. Each such range of sample quantiles was discretized in subintervals of length 0.01 to obtain the grid on which the optimal cut-points are searched.

The function ‘svyglm’ in the ‘survey’ R package has been used again to fit logistic regression models for survey data, and the R packages ‘DiceKriging’ and ‘DiceOptim’ (Roustant et al., 2012) have been used to implement the kriging-based optimization. For each simulation, an initial design of $D = 30$ input vectors was selected using a maximin Latin Hypercube design and was used to start the EGO algorithm to identify the optimal cut-points. The total computing time for the simulations between all three designs was approximately 5 hours.

Summary values of the simulations were calculated to determine the accuracy of the kriging-based optimization method in estimating the true cut-points under each simulation survey design. Firstly, the sample means of the identified optimal cut-points over all 100 simulations and the average of the absolute deviations from these sample means were calculated. The latter is a measure of variability more robust to the influence of potential outliers caused by potential local optima. The EGO algorithm uses a different Latin Hypercube to start a kriging search from each time, but eventually will find the minimum if there is only one. As the EI function can be multimodal, it is possible for it to have local maxima in addition to the global maximum. So, maximizing the EI could get stuck at a local maximum instead of a global maximum. While the EGO algorithm might escape local maxima at the "exploration" stage, this does not always happen. These local maxima then act as outliers in the simulations. In the case that there are multiple minimum values, repeated iterations may help differentiate local versus global minima. While the standard deviation was also reported, it is often sensitive to outliers. Thus, since the average of the absolute deviations is more robust than the standard deviation, it is less affected by potential outliers.

Next, the median of the cut-points over all 100 simulations and their median absolute values, also a measure of spread, were calculated. Note that the median absolute deviation may give unrealistically small variability measurements, as many simulations result in the same identified optimal cut-points. Finally, the average number of EGO iterations over all simulations for each sampling design, rounded up to the nearest integer, was calculated. The summaries of these values are given in Table 5.2 for each

survey design, and histograms displaying the distributions of the identified optimal cut-points for each survey design are displayed in Figures 5.3.4, 5.3.5, and 5.3.6.

Table 5.2: Optimal Cut-Points Simulation Results

Survey Design	**SRS**			**Stratification**			**Clustering**		
	$c_{1,0}$	$c_{2,0}$	$c_{3,0}$	$c_{1,0}$	$c_{2,0}$	$c_{3,0}$	$c_{1,0}$	$c_{2,0}$	$c_{3,0}$
Sample Mean	-0.73	0.05	0.67	0.38	1.25	2.01	-0.74	0.25	0.56
Average Absolute Deviation	0.02	0.07	0.01	0.02	0.03	0.07	0.02	0.03	0.01
Sample Standard Deviation	0.03	0.22	0.02	0.03	0.05	0.01	0.02	0.05	0.02
Sample Median	-0.73	0.08	0.67	0.38	1.26	2.01	-0.74	0.25	0.56
Median Absolute Deviation	0.01	0.02	0.01	0.01	0.01	0.00	0.01	0.03	0.00
EGO Iterations		25			25			22	

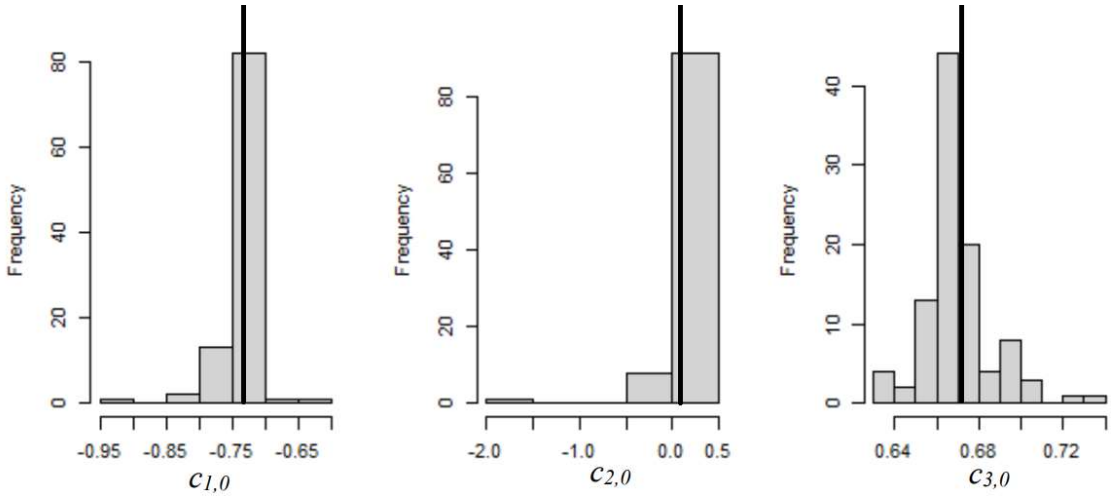

Figure 5.3.4: SRS Simulation Cut-Points Histograms

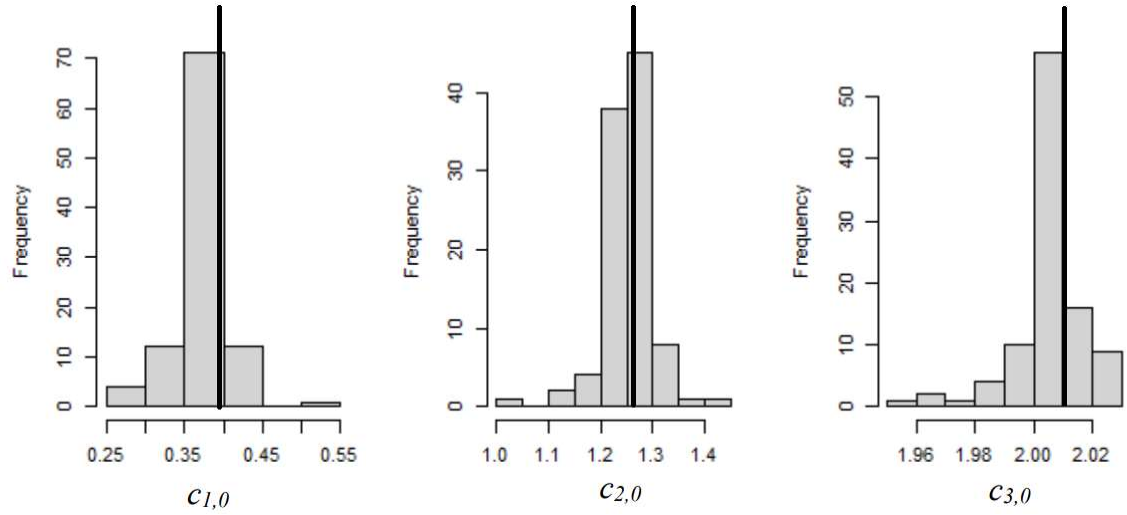

Figure 5.3.5: Stratification Simulation Cut-Points Histograms

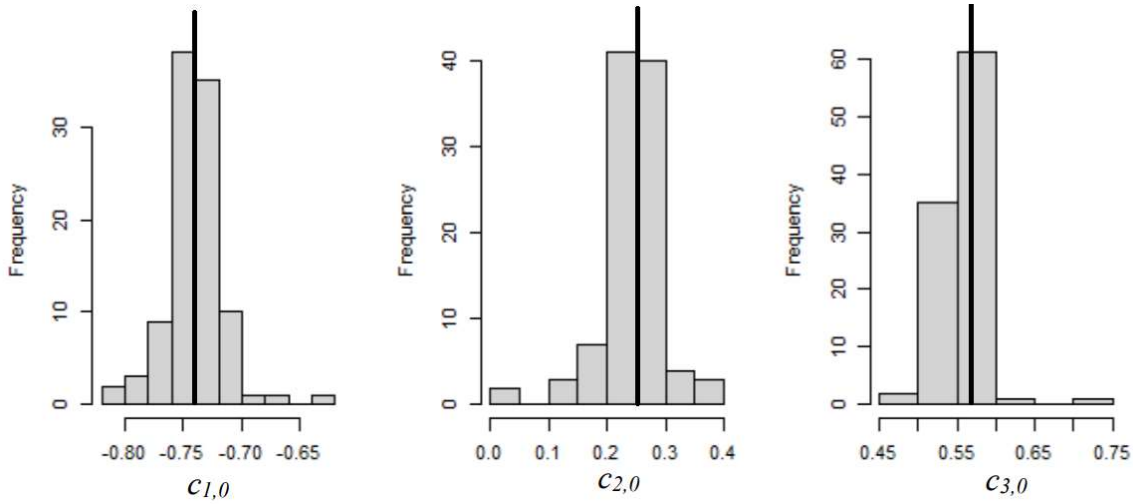

Figure 5.3.6: Clustering Simulation Cut-Points Histograms

For all three survey designs, the sample means and sample medians were close to the true cut-points, while accounting for their uncertainty. This provides empirical evidence that the kriging-based optimization method identifies accurately the optimal cut-points.

5.4 Cardiometabolic Risk Application

Under the NHANES study, the response variable Y for the logistic regression model was defined as 1 if the subject is at cardiometabolic risk and 0 otherwise. The explanatory variables considered were age, BMI, NGS, sex, race/ethnicity, sedentary behavior, poverty status, dietary behavior, and the survey cycle. Three of these explanatory diagnostic variables were included as diagnostic test variables: NGS, BMI (BMXBMI), and age (RIDAGEYR), with a sample size of 1033. These variables were chosen as they were the only statistically significant quantitative explanatory variables (NGS with p-value 0.0432, BMI with p-value < 0.0001, and age with p-value 0.0175) at $\alpha = 0.05$.

A three-dimensional grid of values was created based on each of these explanatory variables. The NGS cut-point was searched in the range of 0.30-0.45, consistent with values found in the literature (Peterson et al., Garcia-Hermoso et al.), by increments of 0.01. The BMI cut-point was searched in the range 17-37, being the rounded values of the 5th and 95th BMI sample quantiles, by increments of 1. Thus, a range of candidate cut-points that excluded subjects with the smallest and largest BMI values was considered. Finally, the age cut-point was searched in the range of 12-24, by increments of 1. These discretized values produce a 16 x 21 x 13 grid of candidate cut-points (c_1, c_2, c_3), for a total of 4,368 combinations. The combination that resulted in the smallest AIC was determined to be the best indicator of cardiometabolic risk.

To demonstrate the accuracy of the proposed method, first each combination of values on the grid was considered by dichotomizing the three diagnostic variables at each

potential cut-point value. Then a logistic regression model for survey data was created including the new dichotomized diagnostic variables, as well as the non-diagnostic explanatory variables. Finally, the AIC value was recorded for each combination. Thus, the AIC values were computed on the entire three-dimensional grid. This direct method found the combination of cut-point values to have the lowest AIC value of 562.0386 at $c_{1,o}$ = 0.38, $c_{2,o}$ = 26.9736, and $c_{3,o}$ = 18. Although this method is fairly accurate, it has a major downfall of being very computationally expensive, taking approximately 3.2 hours of computing time. The direct method on the full grid was also implemented using BIC instead of AIC, identifying the same optimal cut-points for a BIC value of 566.4382 and taking approximately 4.6 hours of computing time.

It is very clear that this direct search method on the grid was not computationally feasible, especially as more diagnostic explanatory variables would be introduced. To address this problem, a kriging-based optimization search was used to find the optimum combination for multiple cut-points.

For the kriging analysis, the same three explanatory diagnostic variables were investigated: NGS, BMI and AGE, therefore k = 3. Note that a diagnostic test based on these three variables is easier to construct than the gold standard Y based on the CMR score, but it may be less accurate than the gold standard to identify subjects at risk. As before, the combination that resulted in the smallest AIC was determined to be the best indicator of cardiometabolic risk. The input space is the 16 x 21 x 13 grid of candidate cut-points (c_1, c_2, c_3) created above. The output is the AIC value at any input (c_1, c_2, c_3), resulting from the logistic regression model that includes the corresponding binary

diagnostic variables while some of the remaining non-diagnostic explanatory variables may be removed during the model selection process.

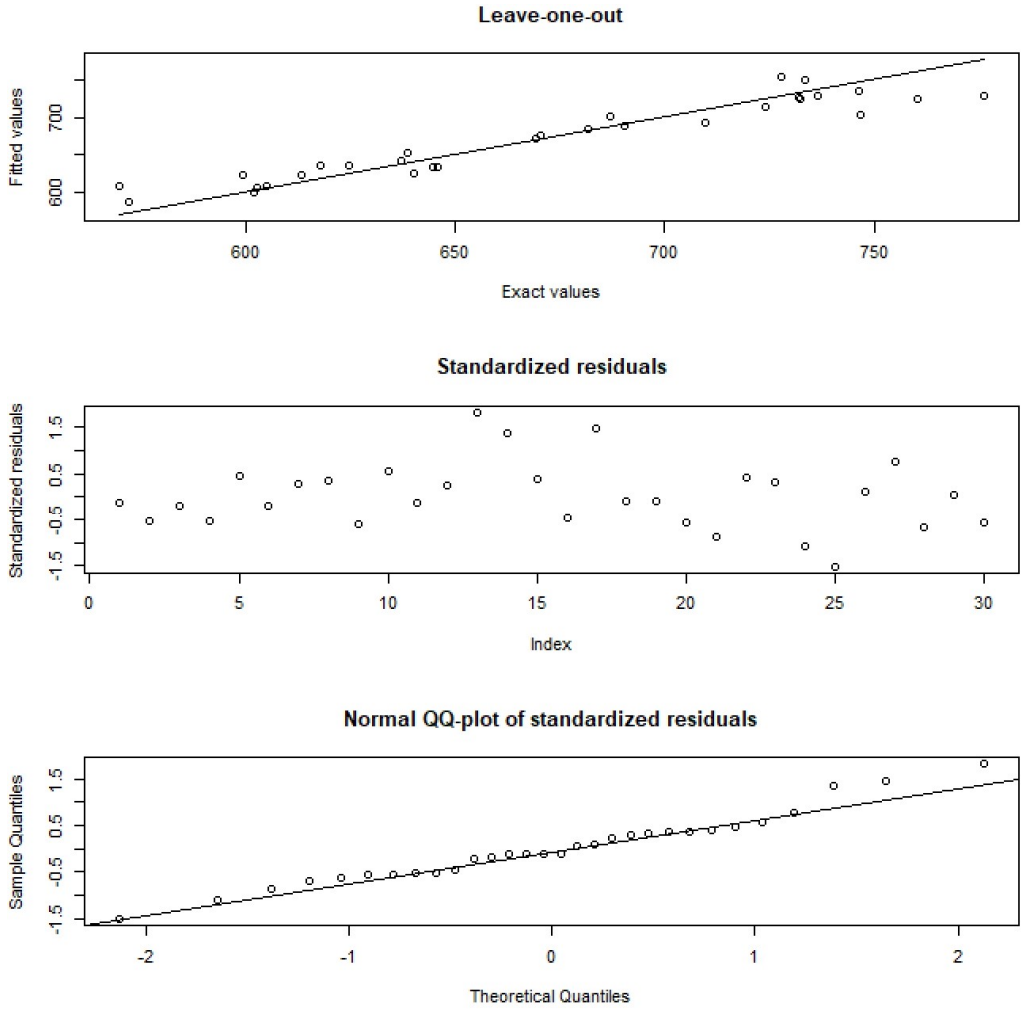

Figure 5.4.1: Leave-one-out cross-validation results of the kriging model

To verify the accuracy and the assumptions of the kriging model, Figure 5.4.1 shows leave-one-out cross-validation results based on D =30 inputs. The upper plot shows a good agreement between the fitted and exact values, where a fitted value is the emulated AIC value at the input left out and the exact value is the actual AIC value at the

same input. The middle plot displays the residuals showing no patterns or outliers, and the lower plot verifies the normality assumption since the QQ-plot is almost linear.

Kriging-based optimization and the EGO algorithm, as described in Sections 5.2.2 and 5.2.3, were implemented to determine the optimal cut-points in a more computationally efficient way. The EGO algorithm used a Latin Hypercube that spread the points out in equal spaces, used kriging to interpolate at any new point, and calculated the AIC at each new point until there was no further significant improvement. While the direct search investigated every possible combination of potential cut-points on the grid, the kriging-based method optimization only looked at certain combinations until the stopping conditions were met. Therefore, there was the possibility of different results at each run of the EGO search due to potential local optima. To come to a conclusion, the kriging search was run 10 times, and the results were compared for each of the 10 runs, each starting with an initial design of $D = 30$ inputs. Figure 5.4.2 shows graphically the results for one of the runs, including the initial design (red points), the optimization path of the EGO algorithm consisting of 8 intermediate iterations (blue points) and the optimal cut-points (green point) at the final iteration. From 10 runs, seven shared the global optimum solution of $(c_{1,0}, c_{2,0}, c_{3,0}) = (0.38, 27, 18)$ with an AIC of 562.0386. Between those seven runs, the average number of iterations was approximately 4.14. The average computing time of the kriging-based optimization method was only 64.51 seconds. For the 10 runs as a whole, the average computing time was 62.12 seconds. Therefore, this method is much more computationally feasible than the direct search, taking approximately one minute using the kriging search and more than three hours using the

direct search. Even with 10 runs, a total computation time of 10-11 minutes is less computationally intensive, which would become more useful with the introduction of more diagnostic variables.

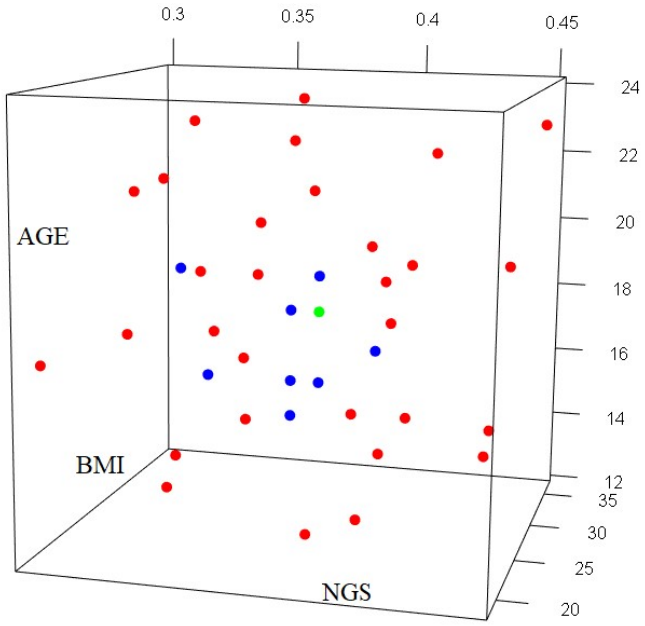

Figure 5.4.2. NHANES EGO Algorithm Example Search

Once the optimal cut-points were determined, the best diagnostic test to be constructed was investigated. This diagnostic test would be used as an indicator of the gold standard (high cardiometabolic score). One diagnostic test would classify a subject as cardiometabolic "high-risk" if NGS $\leq$ 0.38, or BMI $\geq$ 27, or AGE $\geq$ 18. However, it was not clear that this was the best construction of the diagnostic test. Another option might be NGS $\leq$ 0.38, and BMI $\geq$ 27, and AGE $\geq$ 18. Therefore, several combinations for the construction of diagnostic test are possible, as outlined in Table 5.3. The signs of the

corresponding estimated coefficients in Table 4.3 determine the inequalities in the diagnostic tests involving multiple variables.

Table 5.3: Individual Test and Combined Diagnostic Test Descriptions

Combination	Factors
NGS	NGS $\leq$ 0.36
BMI	BMXBMI $\geq$ 27
AGE	RIDAGEYR $\geq$ 21
1	NGSd=1 and BMId=1 and AGEd=1
2	NGSd=1 and (BMId=1 or AGEd=1)
3	BMId=1 and (NGSd=1 or AGEd=1)
4	AGEd=1 and (NGSd=1 or BMId=1)
5	NGSd=1 or BMId=1 or AGEd=1
6	BMId=1 or (NGSd=1 and AGEd=1)
7	NGSd=1 or (BMId=1 and AGEd=1)
8	AGEd=1 or (NGSd=1 and BMId=1)

--- 1-8: where NGSd=1 when NGS $\leq$ 0.38, BMId=1 when BMXBMI $\geq$ 27, AGEd=1 when RIDAGEYR $\geq$ 18

To compare these diagnostic test options, the proportions for at-risk subjects using the diagnostic test and high cardiometabolic score were investigated to determine if there was any overlap in their respective 95% confidence intervals. Then, the whole-sample sensitivity and specificity were calculated for diagnostic tests and high cardiometabolic scores as validation measures. The best diagnostic test was determined to be the one with closer proportions, and higher sensitivity and specificity values. These diagnostic tests were also compared to tests of each diagnostic variable by itself (from Chapter Four), to certify that combining explanatory variables was, in fact, more accurate than basing the cardiometabolic risk status off one variable alone.

Values of estimated proportions as well as the sensitivity and specificity were calculated in R and displayed in Tables 5.4 and 5.5. The 95% confidence interval for the proportion of subjects at risk using the gold standard Y was 0.117 - 0.201.

Table 5.4: Individual Test and Combined Diagnostic Test (DTest) and gold standard (ZBIN) 95% Confidence Intervals

Combination	DTest 95% Confidence Interval	ZBIN 95% Confidence Interval
NGS	0.076 – 0.118	0.117 – 0.201
BMI	0.227 – 0.322	
AGE	0.263 – 0.342	
1	0.040 – 0.085	
2	0.086 – 0.142	
3	0.186 – 0.277	
4	0.149 – 0.228	
5	0.590 – 0.687	
6	0.231 – 0.326	
7	0.205 – 0.301	
8	0.533 – 0.623	

Table 5.5: Individual Test and Combined Diagnostic Test Sensitivity and Specificity Levels

Combination	Sensitivity		Specificity	
	Estimate	**Std. Error**	**Estimate**	**Std. Error**
NGS	0.319	0.033	0.945	0.009
BMI	0.852	0.032	0.835	0.019
AGE	0.451	0.053	0.726	0.022
1	0.268	0.051	0.976	0.008
2	0.411	0.047	0.942	0.010
3	0.790	0.033	0.874	0.015
4	0.647	0.038	0.899	0.014
5	0.955	0.019	0.421	0.030
6	0.852	0.032	0.830	0.018
7	0.804	0.032	0.851	0.016
8	0.879	0.029	0.478	0.029

Among all diagnostic test options, the most optimal test was determined to be combination 3, where BMId=1 and (NGSd=1 or AGEd=1), or equivalently that an individual is "high-risk" if (BMI≥27) and at least one of (AGE≥18), (NGS ≤ 0.38) occurs. This diagnostic test had large sensitivity and specificity values (close to, or above 0.8) and an overlap in the 95% confidence intervals for the proportions of subjects at risk using this diagnostic test and the gold standard, compatible with the two proportions being statistically equal. From the other multi-variable diagnostic tests in Table 5.5, the sensitivity and/or the specificity were lower, and/or the corresponding 95% confidence intervals for the proportion of subjects at risk did not overlap with the gold standard 95% confidence interval.

The best diagnostic test in this application suggested that a subject is at risk if their BMI is at least the optimal cut-point 27, and at least one of the following occurs: their NGS is at most 0.38 or their age is at least 18. This was consistent with the results of the logistic regression model of Y on the explanatory variables discussed above, including the original diagnostic variables. The variable BMI was strongly significant (very small p-value), suggesting that its binary version must be included in the diagnostic test. However, NGS and AGE were only marginally significant (p-values <0.05 but >0.01) suggesting that none of them are important enough to have its binary version absolutely required in the diagnostic test. Rather, including either one of them leads to a better diagnostic test.

The three methods investigated (simulation study, direct search, and kriging-based optimization) to find diagnostic tests indicate that the optimal cut-points were all

accurately identified. In addition, kriging-based optimization has the advantage of being more computationally efficient than the direct search. Finding simple and easily implementable diagnostic tests for medical conditions or diseases is an important problem in health sciences, and having an efficient method to determine this diagnostic test in terms of cardiometabolic risk could be useful in healthcare settings.

5.5 Comparison with Tree-Based Methods

A potential alternative to the proposed logistic regression approach for creating multi-variable diagnostic tests is a tree-based method. To create a decision tree, the predictor space generated by the ranges of explanatory variables is partitioned into regions $R_1, R_2, \ldots, R_n$, also called terminal nodes or leaves. Figure 5.5.1 gives an example of a decision tree partitioned into five leaves. The tree is read from top to bottom, with leaves at the bottom, and internal nodes in between to create splits between the leaves into branches. In the context of creating a diagnostic test, a classification tree is used to predict a qualitative response (such as diseased or non-diseased). At the top of the tree, a split is made based on either a quantitative or qualitative variable. For example, the left branch might signify that an observation is less than or equal to some cut-point value, and the right branch signifies that the observation is greater than the cut-point. This split can also be made on a qualitative category, such as a "Yes" or "No" answer. Starting from the top, the value of each observation is traced downwards until the final leaf is reached. At each leaf, the prediction for the classification is dependent on the most commonly occurring class of observations. For example, if most of the sample in a certain leaf are

considered “diseased,” then any observation in that region will be predicted to be “diseased.”

To choose the cut-points for each internal node, one of the common methods is to minimize the sum of squares of the differences between each observation and the mean response of the observations in the region to which the observation belongs. This method, however, may be too computationally intensive. Other, more efficient methods have been proposed (e.g. James et al., 2021).

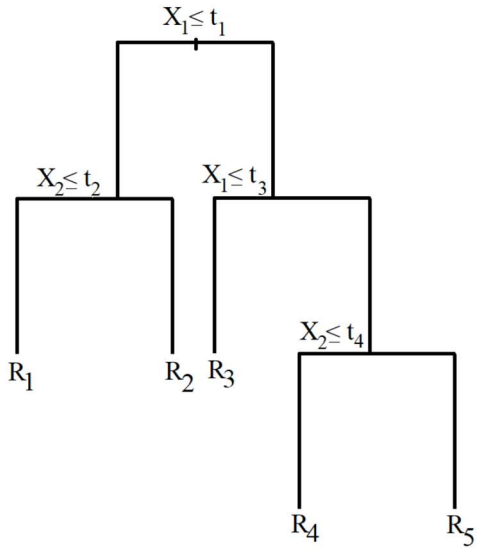

Figure 5.5.1: Decision Tree Example

While useful for classification, the tree-based method has been developed mostly in the context of independent observations, whereas the method proposed in this chapter is applicable with complex survey data. Therefore, the tree-based method described above is not applicable to the complex NHANES survey design considered in this research.

CHAPTER SIX

CONCLUSION

This research has developed statistical methodology for the identification of optimal cut-points for diagnostic variables, in the context of complex survey data. As an application, using the 2011-2012 and 2013-2014 NHANES survey data, a multitude of diagnostic tools have been established as a means of diagnosing cardiometabolic risk.

First, ROC curves for survey data were developed and the cut-points were detected by optimizing appropriate indices, such as the Youden index and upper left index. A replication method for survey data was used to obtain both point estimates and standard errors for these indices. The ROC curves were applied to finding 1-dimensional optimal cut-points for NGS, BMI, and AGE. Both indices identified the same optimal cut-points, specifically 0.45 for NGS, 27 for BMI, and 18 for AGE. To verify the quality of the ROC curves, the area under the curve was calculated for each of the three curves. One of the limitations of the ROC curves for survey data is that it can only be used for a single diagnostic variable. Future areas of improvement for ROC curves in the context of survey data would be implementing covariates as a way of establishing a more accurate curve.

An alternative method for a single diagnostic variable is based on logistic regression for survey data, which can adjust for the effects of covariates, using AIC to select the optimal cut-point. A 1-dimensional simulation study was performed to verify that the proposed method could correctly identify an optimal cut-point based on a single diagnostic variable under logistic regression. Three separate survey designs (simple

random sample, stratified sampling, and two-stage cluster sampling) were considered for the simulation study, which showed that the optimal cut-points identified accurately the true cut-points. Once the simulation study was satisfactory, the logistic regression method was applied separately on three variables of the NHANES data set to determine individual optimal cut-points. Under this method, the optimal cut-points were 0.36 for NGS, 27 for BMI, and 21 for AGE. These results matched for BMI as compared to the ROC method, but not for NGS or AGE. Thus, ROC curves and logistic regression may not always result in the same optimal cut-point for a single diagnostic variable. The results also show that a single variable was determined to not be comprehensive enough to create an accurate diagnostic tool.

Thus, the logistic regression method for survey data was extended to the case of multiple variables to create a multi-dimensional diagnostic test. Although the logistic regression method was shown to be accurate, it was also shown to be time-intensive and computationally inefficient. The kriging-based optimization was proposed as an alternative method to find a more efficient way of constructing a multi-dimensional tool. A simulation study was performed to verify that kriging-based optimization could accurately identify the optimal cut-points. For all three survey designs, the simulation results matched closely the true optimal cut-points. Once this was verified, the kriging-based optimization was applied to the NHANES data set. Under this method, the optimal solution was found to be 0.38 for NGS, 27 for BMI, and 18 for AGE. The proposed method was both accurate and much more computationally efficient. Verification methods involving confidence intervals along with sensitivity and specificity were used

to build an accurate diagnostic test, which resulted in diagnosing an individual as "high-risk" if (BMI ≥ 27) and at least one of (AGE ≥ 18), (NGS ≤ 0.38) occurs. For future work, this diagnostic tool could be expanded to include more diagnostic variables. However, so as to not over-complicate the diagnostic test (as the purpose is to be simple and efficient), there must be a trade-off established between a large-dimensioned, highly accurate diagnostic test and a lower-dimensioned, practical diagnostic test.

Being able to establish an accurate and practical diagnostic test could be highly beneficial in a healthcare setting. Having an efficient method to diagnose patients early could allow for a treatment plan to be established more quickly, thus promoting better health earlier in the process. The methods proposed in this research prove to be useful tools in beginning the search for developing an optimal diagnostic test.

APPENDIX

CODES USED

Chapter 2
Section 2.3
SAS Code for Data Preprocessing: Creating RData9 File

```
/* Import Demographic Data 2011 */
libname in xport '/home/smadi1/DEMO_G.XPT';
data DEMO_G;
set in.DEMO_G;
run;

data DEMO_G;
set DEMO_G;
keep SEQN RIDEXAGM SDMVPSU SDMVSTRA RIAGENDR RIDAGEYR
INDFMPIR RIDRETH3 WTMEC2YR;
run;

proc sort data=DEMO_G;
by SEQN;
run;

/* Add Survey Cycle Variable */
data DEMO_G;
set DEMO_G;
Ind_svy=0;
WTMEC2YR=WTMEC2YR/2;
run;

/* Import Demographic Data 2013 */
libname in xport '/home/smadi1/DEMO_H.XPT';
data DEMO_H;
set in.DEMO_H;
run;

data DEMO_H;
set DEMO_H;
keep SEQN RIDEXAGM SDMVPSU SDMVSTRA RIAGENDR RIDAGEYR
INDFMPIR RIDRETH3 WTMEC2YR;
run;

proc sort data=DEMO_H;
by SEQN;
```

```
run;

/* Add Survey Cycle Variable */
data DEMO_H;
set DEMO_H;
Ind_svy=1;
WTMEC2YR=WTMEC2YR/2;
run;

/* Combine Demographic Data from 2011 and 2013 */
data combineddemo;
 set DEMO_H DEMO_G;
 by SEQN;
run;

/* Import Waist Circumference Data 2011 */
libname in xport '/home/smadi1/BMX_G.XPT';
data BMX_G;
set in.BMX_G;
run;

data BMX_G;
set BMX_G;
keep SEQN BMDBMIC BMXWAIST BMXWT BMXBMI;
run;

proc sort data=BMX_G;
by SEQN;
run;

### … Repeated/Adjusted code for all other listed variables to be uploaded … ###

/* Combine Data Sets Into One File */
data combined;
merge combinedbmx combinedbpq combineddemo combinedbpx combineddiq
combinedglu combinedhdl combinedhsq combinedmcq combinedmgx combinedpaq
combinedtrigly combineddbq;
by SEQN;
run;

/* Define Categorical Age Variable */
data combined;
set combined(where=( RIDAGEYR >= 12 and RIDAGEYR <= 24));
ageY=RIDAGEYR;
```

```
if RIDAGEYR => 12 and RIDAGEYR <= 17 then ageY = 15;
else if RIDAGEYR => 18 and RIDAGEYR <= 24 then ageY = 21;
run;

/* Subset of Data - HSQ Deletions */
data combinedhealthy;
set combined;
   if HSQ500 = 1 then delete;
   if HSQ510 = 1 then delete;
   if HSQ520 = 1 then delete;
run;

/* Changing DBD895 5555 values */
DATA combinedhealthy;
SET combinedhealthy;
if DBD895 = 5555 then DBD895 = 22;
run;

/* Calculate MAP Value */
data combinedhealthy;
set combinedhealthy;
SBP = mean(BPXSY1,BPXSY2,BPXSY3,BPXSY4);
DBP = mean(BPXDI1,BPXDI2,BPXDI3,BPXDI4);
MAP = DBP + (1/3)*(SBP-DBP);
run;

/* Take the maximum gripstrength value of each hand, and determine the highest of the
dominant hand */
DATA combinedhealthy;
SET combinedhealthy;
MAXHAND1 = max(MGXH1T1,MGXH1T2,MGXH1T3);
MAXHAND2 = max(MGXH2T1,MGXH2T2,MGXH2T3);
run;

DATA combinedhealthy;
SET combinedhealthy;
if MGATHAND = 1 and MGD130 = 1 then HANDVALUE = MAXHAND1;
if MGATHAND = 1 and MGD130 = 2 then HANDVALUE = MAXHAND2;
if MGATHAND = 1 and MGD130 = 3 then HANDVALUE =
(mean(MAXHAND1,MAXHAND2));
if MGATHAND = 2 and MGD130 = 1 then HANDVALUE = MAXHAND2;
if MGATHAND = 2 and MGD130 = 2 then HANDVALUE = MAXHAND1;
if MGATHAND = 2 and MGD130 = 3 then HANDVALUE =
(mean(MAXHAND1,MAXHAND2));
```

```
run;

/* Normalized Grip Strength */
DATA combinedhealthy;
SET combinedhealthy;
NGS = HANDVALUE/BMXWT;
run;

/* Delete Missing Values */
data combinedhealthy;
set combinedhealthy;
if BMXWAIST = . then delete;
if LBXTR = . then delete;
if LBDHDD = . then delete;
if LBXGLU = . then delete;
if MAP = . then delete;
if NGS = . then delete;
if PAD680 = . then delete;
if PAD680 = 7777 then delete;
if PAD680 = 9999 then delete;
if BMXBMI = . then delete;
if RIDRETH3 = . then delete;
if RIDAGEYR = . then delete;
if RIAGENDR = . then delete;
if INDFMPIR = . then delete;
if WTMEC2YR = 0 then delete;
if DBD895 = . then delete;
if DBD895 = 7777 then delete;
if DBD895 = 9999 then delete;
run;

/* Take Negative HDL */
DATA combinedhealthy;
SET combinedhealthy;
LBDHDDN = LBDHDD*(-1);
run;

/* Find Mean and SD of Variables */
proc univariate data=combinedhealthy;
  var BMXWAIST;
  var LBXTR;
  var LBDHDDN;
  var LBXGLU;
  var MAP;
```

```
run;

/* Compute Z-Scores of Variables */
DATA combinedhealthy;
SET combinedhealthy;
ZWC = (BMXWAIST-81.3061662)/15.1700266;
ZTRIG = (LBXTR-77.8123324)/52.0690781;
ZHDL = (LBDHDDN-(-53.119303))/11.9473331;
ZGLU = (LBXGLU-94.0871314)/10.9523272;
ZMAP = (MAP-75.1306226)/10.0607979;
run;

/* Z-score = Sum of Individual Z-Scores */
DATA combinedhealthy;
SET combinedhealthy;
ZSCORE = ZWC + ZTRIG + ZHDL + ZGLU + ZMAP;
run;

/* Count Sample Size with a Z-Score */
data combinedzscore;
set combinedhealthy;
   if ZSCORE = . then delete;
run;

/* Check standard deviation of z-score*/
proc univariate data=combinedhealthy;
Title "ZSCORE Data";
  var ZSCORE;
run;

/* Create Response Binary Variable for High Risk */
data combinedhealthy;
set combinedhealthy;
ZBIN=ZSCORE;
if (ZSCORE/2.85328788) => 1 then ZBIN = 1;
else ZBIN = 0;
run;

/* Exporting Data File for R
proc export data=combinedhealthy
(keep=SEQN SDMVSTRA SDMVPSU WTMEC2YR ZSCORE ZBIN
  NGS Ind_svy RIAGENDR RIDAGEYR BMXBMI RIDRETH3 PAD680 INDFMPIR
DBD895)
outfile='/home/smadi1/RData9.txt'
```

```
dbms=tab;
run;
*/
```

Section 2.4.6: Figures 2.4.1 and 2.4.2

```
library(DiceKriging)
library(rgenoud)
t <- seq(from=-2, to=2, length=200)
inputs <- c(-1,-0.5,0,0.5,1)
output <- c(-9,-5,-1,9,11)
theta <- 0.4
sigma <- 5
trend <- c(0,11,2)
model <- km(formula = ~x + I(x^2), design=data.frame(x=inputs), response=output,
covtype="gauss", coef.trend=trend, coef.cov=theta, coef.var=sigma^2)
y <- simulate(model,nsim=5, newdata=data.frame(x=t), cond=TRUE)
ytrend <- trend[1] + trend[2] * t + trend[3] * t^2
X <- expand.grid(x1=seq(0,1,length=4), x2=seq(0,1,length=4))
y <- apply(X,1,branin)
m <- km(~ ., design=X, response=y, covtype="gauss", optim.method="gen",
control=list(trace=FALSE))
n.grid <- 50
x.grid <- seq(0,1,length=n.grid)
X.grid <- expand.grid(x1= x.grid, x2=x.grid)
y.grid <- apply(X.grid, 1, branin)
pred.m <- predict(m, X.grid, "UK")
par(mfrow=c(1,3))
contour(x.grid, x.grid, matrix(y.grid,n.grid,n.grid),50,main="Branin")
points(X[ , 1],X[ , 2], pch=19, cex=1.5, col="red")
contour(x.grid,x.grid,matrix(pred.m$mean, n.grid, n.grid),50,main="Kriging mean")
points(X[ , 1],X[ , 2], pch=19, cex=1.5, col="red")
contour(x.grid,x.grid,matrix(pred.m$sd^2,n.grid, n.grid),15,main="Kriging variance")
points(X[ , 1],X[ , 2], pch=19, cex=1.5, col="red")
plot(m)
```

Section 2.4.6: Figure 2.4.3

```
library(DiceOptim)
library(lhs)
d <- 2
n <- 15
set.seed(0)
```

```
design <- optimumLHS(n,d)
design <- data.frame(design)
name(design) <- c("x1", "x2")
response.branin <- apply(design,1,branin)
fitted.model1 <- km(design=design, response=response.branin)
x.grid <- y.grid <- seq(0,1,length=n.grid <- 25)
design.grid <- expand.grid(x.grid,y.grid)
EI.grid <- apply(design.grid, 1, EI, fitted.model1)
nsteps <- 10
lower <- rep(0,d)
upper <- rep(1,d)
oEGO <- EGO.nsteps(model = fitted.model1, fun=branin, nsteps = nsteps,
lower, upper, control = list(pop.size=20, BFGSburnin=2))
par(mfrow = c(1,2))
response.grid <- apply(design.grid,1,branin)
z.grid <- matrix(response.grid,n.grid,n.grid)
contour(x.grid,y.grid,z.grid,40)
points(design[ ,1], design[ ,2], pch=17, col="blue")
points(oEGO$par, pch=19, col="red")
text(oEGO$par[, 1], oEGO$par[ ,2], labels = 1:nsteps, pos=3)
EI.grid <- apply(design.grid,1,EI,oEGO$lastmodel)
z.grid <- matrix(EI.grid, n.grid, n.grid)
contour(x.grid, y.grid, z.grid, 20)
points(design[, 1], design[, 2], pch=17, col="blue")
points(oEGO$par, pch=19, col="red")
```

Chapter 3

Section 3.3: NGS ROC Curve (repeated/adjusted for BMI and AGE)

```
library(survey);
cardio=read.table("RData9.txt",header=T)
attach(cardio)
# head(cardio, n=5)
NHANES_designD <- svydesign(data=cardio, id=~SDMVPSU, strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
NHANES_objD<-
svyglm(I(ZBIN==1)~NGS+Ind_svy+RIAGENDR+RIDAGEYR+BMXBMI+RIDRETH
3+PAD680+INDFMPIR+DBD895, design = NHANES_designD, family=binomial(link
= "logit"));
summary(NHANES_objD)
race1=rep(0,dim(cardio)[1]);race1[RIDRETH3==1]=1;race1[RIDRETH3==7]=-1;
race2=rep(0,dim(cardio)[1]);race2[RIDRETH3==2]=1;race2[RIDRETH3==7]=-1;
race3=rep(0,dim(cardio)[1]);race3[RIDRETH3==3]=1;race3[RIDRETH3==7]=-1;
```

```
race4=rep(0,dim(cardio)[1]);race4[RIDRETH3==4]=1;race4[RIDRETH3==7]=-1;
race6=rep(0,dim(cardio)[1]);race6[RIDRETH3==6]=1;race6[RIDRETH3==7]=-1;
cardioPlus<-data.frame(cardio,race1,race2,race3,race4,race6)
# head(cardioPlus, n=5)
library(dplyr);

# ---- Taking out ridreth3
cardioPlus2 = select(cardioPlus, -5)
# head(cardioPlus2, n=5)
Ind_SV=rep(0,dim(cardio)[1]);Ind_SV[Ind_svy==0]=1;Ind_SV[Ind_svy==1]=-1;
cardioPlus3<-data.frame(cardioPlus2,Ind_SV)
# head(cardioPlus3)

# ---- Taking out Ind_svy
cardioPlus4 = select(cardioPlus3, -9)
# head(cardioPlus4, n=5)

RIAGENDR2=rep(0,dim(cardio)[1]);RIAGENDR2[RIAGENDR==1]=1;RIAGENDR2[
RIAGENDR==2]=-1;
cardioPlus5<-data.frame(cardioPlus4,RIAGENDR2)
# head(cardioPlus5)
# ---- Taking out RIAGENDR
cardioPlus6 = select(cardioPlus5, -3)
# head(cardioPlus6, n=5)

# ---- Dichotomize NGS at c=cutpoint for Individual Analysis
# ---- NGS > cutpoint = 0 and NGS <= cutpoint = 1
Ng=16;
cCGS=seq(0.30,0.45,,Ng);

SensitNGSArray = rep(0,Ng)
SpecifNGSArray = rep(0,Ng)

YIxArray = rep(0,Ng)
YIyArray = rep(0,Ng)

YIxsdArray = rep(0,Ng)
YIysdArray = rep(0,Ng)

for(i in 1:Ng){
NGSdInd=rep(0,length(NGS));NGSdInd[NGS<=cCGS[i]]=1;
cardioPlusNGSInd<-data.frame(cardioPlus6,NGSdInd)
# head(cardioPlusNGSInd,n=5)
# ---- Taking out NGS
```

```
cardioPlusNGSInd2 = select(cardioPlusNGSInd, -10)
# head(cardioPlusNGSInd2, n=5)

# ---- Set generic data frame for Individual Analysis
dfindngs <- data.frame(cardioPlusNGSInd2)

# ---- Individual Analysis for NGS at cutpoint
DTestNGS=rep(0,length(NGS));DTestNGS[dfindngs$NGSdInd == '1']=1;
cardioPlusTNGS<-data.frame(cardioPlusNGSInd2,DTestNGS)
# head(cardioPlusTNGS,n=5)
NumNGS=rep(0,length(NGS));NumNGS[(ZBIN==1)&(DTestNGS==1)]=1;
DenNGS=rep(0,length(NGS));DenNGS[(ZBIN==1)]=1;
cardioPlusTBNGS<-data.frame(cardioPlusTNGS,NumNGS,DenNGS)
NHANES_designDSensNGS <- svydesign(data=cardioPlusTBNGS, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SensitNGS=svyratio(~NumNGS,~DenNGS,design=NHANES_designDSensNGS)
NumNGS1=rep(0,length(NGS));NumNGS1[(ZBIN==0)&(DTestNGS==0)]=1;
DenNGS1=rep(0,length(NGS));DenNGS1[(ZBIN==0)]=1;
cardioPlusTBNGS1<-data.frame(cardioPlusTNGS,NumNGS1,DenNGS1)
NHANES_designDSpecNGS <- svydesign(data=cardioPlusTBNGS1, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SpecifNGS=svyratio(~NumNGS1,~DenNGS1,design=NHANES_designDSpecNGS)

# ---- Print Sensitivity ratios and SD's (Individual Analysis)
SensitNGSArray[i] = SensitNGS[1]

# ---- Print Specificity ratios and SD's (Individual Analysis)
SpecifNGSArray[i] = SpecifNGS[1]
}

SensitNGSArray = matrix(SensitNGSArray)
SpecifNGSArray = matrix(SpecifNGSArray)
out_df = data.frame(SensitNGSArray, SpecifNGSArray, row.names=cCGS)

#Prints out Matrix of Sensitivity and Specificity Values
print(out_df)
#Plots 1-Specificity vs Sensitivity
plot(1-as.numeric(out_df$SpecifNGSArray), out_df$SensitNGSArray, pch=19, type="b",
xlab="1-Specificity", ylab="Sensitivity")
# Add labels to show cut-points
text(1-as.numeric(out_df$SpecifNGSArray), out_df$SensitNGSArray,
labels=rownames(out_df), pos=1, cex=0.75)
```

```
#Adds points (0,0) and (1,1)
SensitNGSArray01<-rbind(SensitNGSArray,0,1)
SpecifNGSArray01<-rbind(SpecifNGSArray,0,1)

yvalues=as.numeric(SensitNGSArray01)
xvalues=1-as.numeric(SpecifNGSArray01)

yvalues=sort(yvalues)
xvalues=sort(xvalues)

#Adding cut-point labels to the 0 and 1 points
cCGS2<-matrix(cCGS);
cCGS3<-rbind(cCGS2,0,1);
cCGS4<-as.numeric(cCGS3);
cCGS5<-sort(cCGS4);

out_df2 = data.frame(xvalues, yvalues, row.names=cCGS5)
plot(out_df2$xvalues, out_df2$yvalues, pch=19, type="b", xlab="1-Specificity",
ylab="Sensitivity", main="ROC Curve for NGS Cut-points")
text(out_df2$xvalues, out_df2$yvalues, labels=rownames(out_df2), pos=4, cex=0.75)

#Centered (Trapezoidal) Riemann Sum
library(pracma);
RiemannSumTrap <- trapz(xvalues,yvalues)
RiemannSumTrap
# 0.7116769

#Sets up Right and Left Riemann Sums
sum_riemannright = 0
sum_riemannleft = 0

for (i in 1:Ng+2) {
 if (i < Ng+2){
   diffmax = abs(xvalues[i] - xvalues[i+1])
   max_y = max(yvalues[i], yvalues[i+1])
   sum_riemannright = sum_riemannright + diffmax * max_y
 }}

for (i in 1:Ng+2) {
 if (i < Ng+2){
   diffmin = abs(xvalues[i] - xvalues[i+1])
   min_y = min(yvalues[i], yvalues[i+1])
   sum_riemannleft = sum_riemannleft + diffmin * min_y
```

```
 }}

#print(sum_riemannright)
#print(sum_riemannleft)

#Calculates bound on error, which is at most |Rn-Ln|
Rn=as.numeric(sum_riemannright);
Ln=as.numeric(sum_riemannleft);
abs(Rn-Ln)
#0.3160465

#Upper Left Index
ULone=1-as.numeric(SensitNGSArray)
ULtwo=1-as.numeric(SpecifNGSArray)
ULindex=ULone^2+ULtwo^2
which.min(ULindex)
#16
ULindex[16]
#0.2158827
cCGS[16]
#0.45

#Youden Index
YI1=as.numeric(SensitNGSArray)
YI2=as.numeric(SpecifNGSArray)
YI=YI1+YI2-1
which.max(YI)
#16
YI[16]
#0.3661154
cCGS[16]
#0.45
```

NGS ROC Youden Index Replication Method
(Repeated for Upper Left Index, repeated/adjusted for BMI and AGE)

```
rm(list=ls())
library(survey)
cardio=read.table("RData9.txt",header=T)
attach(cardio)
# head(cardio, n=5)
NHANES_designD <- svydesign(data=cardio, id=~SDMVPSU, strata=~SDMVSTRA,
weights=~WTMEC2YR,
```

```
nest=TRUE)
NHANES_objD<-
svyglm(I(ZBIN==1)~NGS+Ind_svy+RIAGENDR+RIDAGEYR+BMXBMI+RIDRETH
3+PAD680+INDFMPIR+DBD895, design = NHANES_designD, family=binomial(link
= "logit"))
summary(NHANES_objD)
race1=rep(0,dim(cardio)[1]);race1[RIDRETH3==1]=1;race1[RIDRETH3==7]=-1;
race2=rep(0,dim(cardio)[1]);race2[RIDRETH3==2]=1;race2[RIDRETH3==7]=-1;
race3=rep(0,dim(cardio)[1]);race3[RIDRETH3==3]=1;race3[RIDRETH3==7]=-1;
race4=rep(0,dim(cardio)[1]);race4[RIDRETH3==4]=1;race4[RIDRETH3==7]=-1;
race6=rep(0,dim(cardio)[1]);race6[RIDRETH3==6]=1;race6[RIDRETH3==7]=-1;
cardioPlus<-data.frame(cardio,race1,race2,race3,race4,race6)
# head(cardioPlus, n=5)
library(dplyr)
# ---- Taking out ridreth3
cardioPlus2 = select(cardioPlus, -5)
# head(cardioPlus2, n=5)
Ind_SV=rep(0,dim(cardio)[1]);Ind_SV[Ind_svy==0]=1;Ind_SV[Ind_svy==1]=-1;
cardioPlus3<-data.frame(cardioPlus2,Ind_SV)
# head(cardioPlus3)
# ---- Taking out Ind_svy
cardioPlus4 = select(cardioPlus3, -9)
# head(cardioPlus4, n=5)

RIAGENDR2=rep(0,dim(cardio)[1]);RIAGENDR2[RIAGENDR==1]=1;RIAGENDR2[
RIAGENDR==2]=-1;
cardioPlus5<-data.frame(cardioPlus4,RIAGENDR2)
# head(cardioPlus5)
# ---- Taking out RIAGENDR
cardioPlus6 = select(cardioPlus5, -3)
# head(cardioPlus6, n=5)

# ---- Dichotomize NGS at c=cutpoint for Individual Analysis
# ---- NGS > cutpoint = 0 and NGS <= cutpoint = 1
Ng=16;
cCGS=seq(0.30,0.45,,Ng);
SensitNGSArray = rep(0,Ng)
SpecifNGSArray = rep(0,Ng)

for(i in 1:Ng){
NGSdInd=rep(0,length(NGS));NGSdInd[NGS<=cCGS[i]]=1;
cardioPlusNGSInd<-data.frame(cardioPlus6,NGSdInd)
# head(cardioPlusNGSInd,n=5)
# ---- Taking out NGS
```

```
cardioPlusNGSInd2 = select(cardioPlusNGSInd, -10)
# head(cardioPlusNGSInd2, n=5)

# ---- Set generic data frame for Individual Analysis
dfindNGS <- data.frame(cardioPlusNGSInd2)

# ---- Individual Analysis for NGS at cutpoint
DTestNGS=rep(0,length(NGS));DTestNGS[dfindNGS$NGSdInd == '1']=1;
cardioPlusTNGS<-data.frame(cardioPlusNGSInd2,DTestNGS)
# head(cardioPlusTNGS,n=5)
NumNGS=rep(0,length(NGS));NumNGS[(ZBIN==1)&(DTestNGS==1)]=1;
DenNGS=rep(0,length(NGS));DenNGS[(ZBIN==1)]=1;
cardioPlusTBNGS<-data.frame(cardioPlusTNGS,NumNGS,DenNGS)
NHANES_designDSensNGS <- svydesign(data=cardioPlusTBNGS, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SensitNGS=svyratio(~NumNGS,~DenNGS,design=NHANES_designDSensNGS)
NumNGS1=rep(0,length(NGS));NumNGS1[(ZBIN==0)&(DTestNGS==0)]=1;
DenNGS1=rep(0,length(NGS));DenNGS1[(ZBIN==0)]=1;
cardioPlusTBNGS1<-data.frame(cardioPlusTNGS,NumNGS1,DenNGS1)
NHANES_designDSpecNGS <- svydesign(data=cardioPlusTBNGS1, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SpecifNGS=svyratio(~NumNGS1,~DenNGS1,design=NHANES_designDSpecNGS)

# ---- Print Sensitivity ratios and SD's (Individual Analysis)
SensitNGSArray[i] = SensitNGS[1]

# ---- Print Specificity ratios and SD's (Individual Analysis)
SpecifNGSArray[i] = SpecifNGS[1]
}

#Youden Index
YI1=as.numeric(SensitNGSArray)
YI2=as.numeric(SpecifNGSArray)
YI=YI1+YI2-1
optim.index=which.max(YI)

NGSdInd=rep(0,length(NGS));NGSdInd[NGS<=cCGS[optim.index]]=1;
cardioPlusNGSInd<-data.frame(cardioPlus6,NGSdInd)
# head(cardioPlusNGSInd,n=5)
# ---- Taking out NGS
cardioPlusNGSInd2 = select(cardioPlusNGSInd, -10)
# head(cardioPlusNGSInd2, n=5)
```

```
# ---- Set generic data frame for Individual Analysis
dfindNGS <- data.frame(cardioPlusNGSInd2)

# ---- Individual Analysis for NGS at cutpoint
DTestNGS=rep(0,length(NGS));DTestNGS[dfindNGS$NGSdInd == '1']=1;
cardioPlusTNGS<-data.frame(cardioPlusNGSInd2,DTestNGS)
# head(cardioPlusTNGS,n=5)
NumNGS=rep(0,length(NGS));NumNGS[(ZBIN==1)&(DTestNGS==1)]=1;
DenNGS=rep(0,length(NGS));DenNGS[(ZBIN==1)]=1;
cardioPlusTBNGS<-data.frame(cardioPlusTNGS,NumNGS,DenNGS)
NHANES_designDSensNGS <- svydesign(data=cardioPlusTBNGS, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SensitNGS=svyratio(~NumNGS,~DenNGS,design=NHANES_designDSensNGS)
print(SensitNGS)
#ratios = 0.5965191, SE = 0.05315409

NHANES_designDSensNGS_rep <- as.svrepdesign(NHANES_designDSensNGS)
SensitR<-function(WTMEC2YR,cardioPlusTBNGS) {
sum(WTMEC2YR*NumNGS)/sum(WTMEC2YR*DenNGS)
}
print(withReplicates(NHANES_designDSensNGS_rep, SensitR))
#theta = 0.59652, SE = 0.0534
### same results as SensitNGS, but using replications such as jackknife

NumNGS1=rep(0,length(NGS));NumNGS1[(ZBIN==0)&(DTestNGS==0)]=1;
DenNGS1=rep(0,length(NGS));DenNGS1[(ZBIN==0)]=1;
cardioPlusTBNGS1<-data.frame(cardioPlusTNGS,NumNGS1,DenNGS1)
NHANES_designDSpecNGS <- svydesign(data=cardioPlusTBNGS1, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SpecifNGS=svyratio(~NumNGS1,~DenNGS1,design=NHANES_designDSpecNGS)
print(SpecifNGS)
# ratios = 0.7695963, SE = 0.01783795

NHANES_designDSpecNGS_rep <- as.svrepdesign(NHANES_designDSpecNGS)
SpecifR<-function(WTMEC2YR,cardioPlusTBNGS1) {
sum(WTMEC2YR*NumNGS1)/sum(WTMEC2YR*DenNGS1)
}
print(withReplicates(NHANES_designDSpecNGS_rep, SpecifR))
#theta= 0.7696, SE = 0.0178
### same results as SpecifNGS, but using replications such as jackknife
```

```
################ This uses the replication method to find SE for the estimator of
Youden index
cardioPlusTBNGSY<-
data.frame(cardioPlusTNGS,NumNGS1,DenNGS1,NumNGS,DenNGS)
NHANES_designDSpecNGSY <- svydesign(data=cardioPlusTBNGSY,
id=~SDMVPSU, strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
NHANES_designDSpecNGSY_rep <- as.svrepdesign(NHANES_designDSpecNGSY)

#NHANES_designDSpecNGSY_rep <-
as.svrepdesign(NHANES_designDSpecNGSY,type="bootstrap")
Youd<-function(WTMEC2YR,cardioPlusTBNGSY) {
sum(WTMEC2YR*NumNGS)/sum(WTMEC2YR*DenNGS)+
sum(WTMEC2YR*NumNGS1)/sum(WTMEC2YR*DenNGS1) - 1
}
print(withReplicates(NHANES_designDSpecNGSY_rep, Youd))
#theta= 0.36612, SE = 0.0586
```

Table 3.5

```
library(survey)
cardio=read.table("RData9.txt",header=T)
attach(cardio)
# head(cardio, n=5)
NHANES_designD <- svydesign(data=cardio, id=~SDMVPSU, strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
NHANES_objD<-
svyglm(I(ZBIN==1)~NGS+Ind_svy+RIAGENDR+RIDAGEYR+BMXBMI+RIDRETH
3+PAD680+INDFMPIR+DBD895, design = NHANES_designD, family=binomial(link
= "logit"))
summary(NHANES_objD)
race1=rep(0,dim(cardio)[1]);race1[RIDRETH3==1]=1;race1[RIDRETH3==7]=-1;
race2=rep(0,dim(cardio)[1]);race2[RIDRETH3==2]=1;race2[RIDRETH3==7]=-1;
race3=rep(0,dim(cardio)[1]);race3[RIDRETH3==3]=1;race3[RIDRETH3==7]=-1;
race4=rep(0,dim(cardio)[1]);race4[RIDRETH3==4]=1;race4[RIDRETH3==7]=-1;
race6=rep(0,dim(cardio)[1]);race6[RIDRETH3==6]=1;race6[RIDRETH3==7]=-1;
cardioPlus<-data.frame(cardio,race1,race2,race3,race4,race6)
# head(cardioPlus, n=5)
library(dplyr)

# ---- Taking out ridreth3
cardioPlus2 = select(cardioPlus, -5)
# head(cardioPlus2, n=5)
```

```
Ind_SV=rep(0,dim(cardio)[1]);Ind_SV[Ind_svy==0]=1;Ind_SV[Ind_svy==1]=-1;
cardioPlus3<-data.frame(cardioPlus2,Ind_SV)
# head(cardioPlus3)

# ---- Taking out Ind_svy
cardioPlus4 = select(cardioPlus3, -9)
# head(cardioPlus4, n=5)

RIAGENDR2=rep(0,dim(cardio)[1]);RIAGENDR2[RIAGENDR==1]=1;RIAGENDR2[
RIAGENDR==2]=-1;
cardioPlus5<-data.frame(cardioPlus4,RIAGENDR2)
# head(cardioPlus5)
# ---- Taking out RIAGENDR
cardioPlus6 = select(cardioPlus5, -3)
# head(cardioPlus6, n=5)


# ---- Dichotomize NGS at c=0.45 for Individual Analysis
# ---- NGS > 0.45 = 0 and NGS <= 0.45 = 1
NGSdInd=rep(0,length(NGS));NGSdInd[NGS<=0.45]=1;
cardioPlusNGSInd<-data.frame(cardioPlus6,NGSdInd)
# head(cardioPlusNGSInd,n=5)
# ---- Taking out NGS
cardioPlusNGSInd2 = select(cardioPlusNGSInd, -10)
# head(cardioPlusNGSInd2, n=5)

# ---- Dichotomize BMI at c=27 for Individual Analysis
# ---- BMI < 27 = 0 and BMI >= 27 = 1
BMIdInd=rep(0,length(BMXBMI));BMIdInd[BMXBMI>=27]=1;
cardioPlusBMIInd<-data.frame(cardioPlus6,BMIdInd)
# head(cardioPlusBMIInd,n=5)
# ---- Taking out BMXBMI
cardioPlusBMIInd2 = select(cardioPlusBMIInd, -2)
# head(cardioPlusBMIInd2, n=5)

# ---- Dichotomize AGE at c=18 for Individual Analysis
# ---- AGE < 18 = 0 and AGE >= 18 = 1
AGEdInd=rep(0,length(RIDAGEYR));AGEdInd[RIDAGEYR>=18]=1;
cardioPlusAGEInd<-data.frame(cardioPlus6,AGEdInd)
# head(cardioPlusAGEInd,n=5)
# ---- Taking out RIDAGEYR
cardioPlusAGEInd2 = select(cardioPlusAGEInd, -3)
# head(cardioPlusAGEInd2, n=5)
```

```
# ---- Set generic data frame for Individual Analysis
dfindngs <- data.frame(cardioPlusNGSInd2)
dfindbmi <- data.frame(cardioPlusBMIInd2)
dfindage <- data.frame(cardioPlusAGEInd2)

# ---- Individual Analysis for NGS at c=0.45
DTestNGS=rep(0,length(NGS));DTestNGS[dfindngs$NGSdInd == '1']=1;
cardioPlusTNGS<-data.frame(cardioPlusNGSInd2,DTestNGS)
# head(cardioPlusTNGS,n=5)
NumNGS=rep(0,length(NGS));NumNGS[(ZBIN==1)&(DTestNGS==1)]=1;
DenNGS=rep(0,length(NGS));DenNGS[(ZBIN==1)]=1;
cardioPlusTBNGS<-data.frame(cardioPlusTNGS,NumNGS,DenNGS)
NHANES_designDSensNGS <- svydesign(data=cardioPlusTBNGS, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SensitNGS=svyratio(~NumNGS,~DenNGS,design=NHANES_designDSensNGS)
NumNGS1=rep(0,length(NGS));NumNGS1[(ZBIN==0)&(DTestNGS==0)]=1;
DenNGS1=rep(0,length(NGS));DenNGS1[(ZBIN==0)]=1;
cardioPlusTBNGS1<-data.frame(cardioPlusTNGS,NumNGS1,DenNGS1)
NHANES_designDSpecNGS <- svydesign(data=cardioPlusTBNGS1, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SpecifNGS=svyratio(~NumNGS1,~DenNGS1,design=NHANES_designDSpecNGS)
NHANES_designDNGS <- svydesign(data=cardioPlusTNGS, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTestNGSmean = svymean(DTestNGS,design=NHANES_designDNGS)
ZBINNGSmean = svymean(ZBIN,design=NHANES_designDNGS)


# ---- Individual Analysis for BMI at c=27
DTestBMI=rep(0,length(BMXBMI));DTestBMI[dfindbmi$BMIdInd == '1']=1;
cardioPlusTBMI<-data.frame(cardioPlusBMIInd2,DTestBMI)
# head(cardioPlusTBMI,n=5)
NumBMI=rep(0,length(BMXBMI));NumBMI[(ZBIN==1)&(DTestBMI==1)]=1;
DenBMI=rep(0,length(BMXBMI));DenBMI[(ZBIN==1)]=1;
cardioPlusTBBMI<-data.frame(cardioPlusTBMI,NumBMI,DenBMI)
NHANES_designDSensBMI <- svydesign(data=cardioPlusTBBMI, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SensitBMI=svyratio(~NumBMI,~DenBMI,design=NHANES_designDSensBMI)
NumBMI1=rep(0,length(BMXBMI));NumBMI1[(ZBIN==0)&(DTestBMI==0)]=1;
DenBMI1=rep(0,length(BMXBMI));DenBMI1[(ZBIN==0)]=1;
cardioPlusTBBMI1<-data.frame(cardioPlusTBMI,NumBMI1,DenBMI1)
```

```
NHANES_designDSpecBMI <- svydesign(data=cardioPlusTBBMI1, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SpecifBMI=svyratio(~NumBMI1,~DenBMI1,design=NHANES_designDSpecBMI)
NHANES_designDBMI <- svydesign(data=cardioPlusTBMI, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTestBMImean = svymean(DTestBMI,design=NHANES_designDBMI)
ZBINBMImean = svymean(ZBIN,design=NHANES_designDBMI)

# ---- Individual Analysis for AGE at c=18
DTestAGE=rep(0,length(RIDAGEYR));DTestAGE[dfindage$AGEdInd == '1']=1;
cardioPlusTAGE<-data.frame(cardioPlusAGEInd2,DTestAGE)
# head(cardioPlusTAGE,n=5)
NumAGE=rep(0,length(RIDAGEYR));NumAGE[(ZBIN==1)&(DTestAGE==1)]=1;
DenAGE=rep(0,length(RIDAGEYR));DenAGE[(ZBIN==1)]=1;
cardioPlusTBAGE<-data.frame(cardioPlusTAGE,NumAGE,DenAGE)
NHANES_designDSensAGE <- svydesign(data=cardioPlusTBAGE, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SensitAGE=svyratio(~NumAGE,~DenAGE,design=NHANES_designDSensAGE)
NumAGE1=rep(0,length(RIDAGEYR));NumAGE1[(ZBIN==0)&(DTestAGE==0)]=1;
DenAGE1=rep(0,length(RIDAGEYR));DenAGE1[(ZBIN==0)]=1;
cardioPlusTBAGE1<-data.frame(cardioPlusTAGE,NumAGE1,DenAGE1)
NHANES_designDSpecAGE <- svydesign(data=cardioPlusTBAGE1, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SpecifAGE=svyratio(~NumAGE1,~DenAGE1,design=NHANES_designDSpecAGE)
NHANES_designDAGE <- svydesign(data=cardioPlusTAGE, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTestAGEmean = svymean(DTestAGE,design=NHANES_designDAGE)
ZBINAGEmean = svymean(ZBIN,design=NHANES_designDAGE)


# ---- CI's for Individual Analysis
df = 61-29
CINGSlower = mean(DTestNGSmean) - qt(0.975,df)*SE(DTestNGSmean)
CINGSupper = mean(DTestNGSmean) + qt(0.975,df)*SE(DTestNGSmean)
CIBMIlower = mean(DTestBMImean) - qt(0.975,df)*SE(DTestBMImean)
CIBMIupper = mean(DTestBMImean) + qt(0.975,df)*SE(DTestBMImean)
CIAGElower = mean(DTestAGEmean) - qt(0.975,df)*SE(DTestAGEmean)
CIAGEupper = mean(DTestAGEmean) + qt(0.975,df)*SE(DTestAGEmean)
```

```
CINGSlower;CINGSupper
#0.2484499, 0.3287593
CIBMIlower;CIBMIupper
#0.2265155, 0.3221765
CIAGElower;CIAGEupper
#0.4877774, 0.5741275

## same ZBIN Confidence Interval from Section 4.4 ##
```

Chapter 4

Section 4.3: Clustering 1-Dimensional Simulation Study
(Repeated/adjusted for SRS and Stratification)

```
rm(list=ls())
library(survey)
library(MASS)

################     LOGISTIC SIMULATIONS  ###################

bet0=-5;bet=c(4, -2, 3, 0.1, -0.01);Rsig=1;
N=1000;Nrep=100;

# define the clustering variable and weights
Npsu=80;npsu=8;Mi=1250;mi=125;N=npsu*mi;
clust<-rep(kronecker(1:npsu,rep(1,mi)))
Wghts<-(Npsu/npsu)*(Mi/mi) *rep(1,N);

# equal-correlation (compound-symmetry) within each cluster
Rsig=1;
X=matrix(Rsig*rnorm(N*length(bet)),N,length(bet));
Cr=matrix(.2,mi,mi);diag(Cr)=rep(1,mi);
for (i in 1:length(bet)){X[,i]=kronecker(diag(npsu),t(chol(Cr)))%*%X[,i]}

Lo1=round(quantile(X[,1],.01),2);Up1=round(quantile(X[,1],.99),2);
design=seq(Lo1,Up1,.01);N1=length(design);

#########     SIMUL CUT-POINTS

Xc=X;
X1=rep(0,N);C1=round(quantile(X[,1],.25),2);X1[X[,1]>=C1]=1;Xc[,1]=X1;

Probs=exp(bet0+Xc%*%bet)/(1+exp(bet0+Xc%*%bet));
```

```
RezSim=matrix(NA,Nrep,2)

for (rep in 1:Nrep){
Y=rep(NA,N);for (i in 1:N){Y[i]=rbinom(1, 1, Probs[i])}

response.AIC=rep(NA,N1);
for (i in 1:N1){
Xd1=rep(0,N);Xd1[X[,1]>=design[i]]=1;
Xd2=X[,2];Xd3=X[,3];Xd4=X[,4];Xd5=X[,5];

Dat_clusD=data.frame(Xd1,Xd2,Xd3,Xd4,Xd5,Y,Wghts);
Lro_design <- svydesign(data=Dat_clusD, id=~clust, weights=~Wghts, nest=TRUE)
LroD=svyglm(I(Y==1)~Xd1+Xd2+Xd3+Xd4+Xd5,design =
Lro_design,family=binomial(link = "logit"))
Lro_stepD<-stepAIC(LroD, trace = FALSE, scope = list(lower = ~ Xd1, upper = LroD))
response.AIC[i]=AIC(Lro_stepD)[2]
}

OptPt=(1:N1)[response.AIC==min(response.AIC)];OptPt=OptPt[1]
RezSim[rep,]=c(design[OptPt], response.AIC[OptPt])
print(rep)
}

### plots AIC at the last iteration
plot(design,response.AIC)

dev.new();par(mfrow=c(1,2));
hist(RezSim[,1]);hist(RezSim[,2]);

SDm<-function(x){mean(abs(x-mean(x)))}
print(C1)
print(c(apply(RezSim,2,mean)))
print(c(apply(RezSim,2,SDm)))
# Standard deviation for all simulations
print(apply(RezSim,2,sd))
# Median values of the simulations
print(apply(RezSim,2,median))
# Median absolue deviation
print(apply(RezSim,2,mad))
dev.new();par(mfrow=c(2,3));
hist(RezSim[,1]);hist(RezSim[,2]);

system.time(source("ClusteringSim1Dim.R"))
```

Section 4.4: NGS 1-Dimensional Loop
(Repeated/adjusted for BMI and AGE)

```
library(survey)
cardio=read.table("RData9.txt",header=T)
attach(cardio)
head(cardio, n=10)
race1=rep(0,dim(cardio)[1]);race1[RIDRETH3==1]=1;race1[RIDRETH3==7]=-1;
race2=rep(0,dim(cardio)[1]);race2[RIDRETH3==2]=1;race2[RIDRETH3==7]=-1;
race3=rep(0,dim(cardio)[1]);race3[RIDRETH3==3]=1;race3[RIDRETH3==7]=-1;
race4=rep(0,dim(cardio)[1]);race4[RIDRETH3==4]=1;race4[RIDRETH3==7]=-1;
race6=rep(0,dim(cardio)[1]);race6[RIDRETH3==6]=1;race6[RIDRETH3==7]=-1;
RIAGENDR2=rep(0,dim(cardio)[1]);RIAGENDR2[RIAGENDR==1]=1;RIAGENDR2[
RIAGENDR==2]=-1;
Ind_SV=Ind_svy;Ind_SV[Ind_svy==0]=1;Ind_SV[Ind_svy==1]=-1;

cardioPlus<-data.frame(cardio,race1,race2,race3,race4,race6,RIAGENDR2,Ind_SV)

library(lhs)
library(DiceKriging)
library(DiceOptim)
library(knitr)
library(rgl)
library(MASS)

Ng=16;
cCGS=seq(0.30,0.45,,Ng);
AICval=rep(NA,Ng)

for (i in 1:Ng){
        NGSd2=rep(0,length(NGS));NGSd2[NGS>=cCGS[i]]=1;
        cardioPlusD<-data.frame(cardioPlus,NGSd2)
        NHANES_designD4 <- svydesign(data=cardioPlusD, id=~SDMVPSU,
strata=~SDMVSTRA,          weights=~WTMEC2YR, nest=TRUE)
        NHANES_objD4<-
        suppressWarnings(svyglm(I(ZBIN==1)~NGSd2+Ind_SV+RIDAGEYR+BMXB
MI+RIAGENDR2+race1+
        race2+race3+race4+race6+PAD680+INDFMPIR+DBD895, design =
NHANES_designD4, family=binomial(link = "logit")))
        NHANES_step2<-suppressWarnings(stepAIC(NHANES_objD4, trace = FALSE,
scope = list(lower = ~ NGSd2, upper = NHANES_objD4)))
        AICval[i]= suppressWarnings(AIC(NHANES_step2)[2])
}
```

```
plot(cCGS,AICval,type="b")
cCGS
AICval
min(AICval)
which(AICval == min(AICval), arr.ind=TRUE)
```

R Code for Individual and Combination Confidence Intervals and Sensitivity/Specificity Levels (Sections 4.3, 4.4, and 5.4)

```
library(survey)
cardio=read.table("RData9.txt",header=T)
attach(cardio)
# head(cardio, n=5)
NHANES_designD <- svydesign(data=cardio, id=~SDMVPSU, strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
NHANES_objD<-
svyglm(I(ZBIN==1)~NGS+Ind_svy+RIAGENDR+RIDAGEYR+BMXBMI+RIDRETH
3+PAD680+INDFMPIR+DBD895, design = NHANES_designD, family=binomial(link
= "logit"))
summary(NHANES_objD)
race1=rep(0,dim(cardio)[1]);race1[RIDRETH3==1]=1;race1[RIDRETH3==7]=-1;
race2=rep(0,dim(cardio)[1]);race2[RIDRETH3==2]=1;race2[RIDRETH3==7]=-1;
race3=rep(0,dim(cardio)[1]);race3[RIDRETH3==3]=1;race3[RIDRETH3==7]=-1;
race4=rep(0,dim(cardio)[1]);race4[RIDRETH3==4]=1;race4[RIDRETH3==7]=-1;
race6=rep(0,dim(cardio)[1]);race6[RIDRETH3==6]=1;race6[RIDRETH3==7]=-1;
cardioPlus<-data.frame(cardio,race1,race2,race3,race4,race6)
# head(cardioPlus, n=5)
library(dplyr)

# ---- Taking out ridreth3
cardioPlus2 = select(cardioPlus, -5)
# head(cardioPlus2, n=5)
Ind_SV=rep(0,dim(cardio)[1]);Ind_SV[Ind_svy==0]=1;Ind_SV[Ind_svy==1]=-1;
cardioPlus3<-data.frame(cardioPlus2,Ind_SV)
# head(cardioPlus3)

# ---- Taking out Ind_svy
cardioPlus4 = select(cardioPlus3, -9)
# head(cardioPlus4, n=5)

RIAGENDR2=rep(0,dim(cardio)[1]);RIAGENDR2[RIAGENDR==1]=1;RIAGENDR2[
RIAGENDR==2]=-1;
cardioPlus5<-data.frame(cardioPlus4,RIAGENDR2)
```

```
# head(cardioPlus5)
# ---- Taking out RIAGENDR
cardioPlus6 = select(cardioPlus5, -3)
# head(cardioPlus6, n=5)

# ---- Dichotomize NGS at c=0.36 for Individual Analysis
# ---- NGS > 0.36 = 0 and NGS <= 0.36 = 1
NGSdInd=rep(0,length(NGS));NGSdInd[NGS<=0.36]=1;
cardioPlusNGSInd<-data.frame(cardioPlus6,NGSdInd)
# head(cardioPlusNGSInd,n=5)
# ---- Taking out NGS
cardioPlusNGSInd2 = select(cardioPlusNGSInd, -10)
# head(cardioPlusNGSInd2, n=5)

# ---- Dichotomize BMI at c=27 for Individual Analysis
# ---- BMI < 27 = 0 and BMI >= 27 = 1
BMIdInd=rep(0,length(BMXBMI));BMIdInd[BMXBMI>=27]=1;
cardioPlusBMIInd<-data.frame(cardioPlus6,BMIdInd)
# head(cardioPlusBMIInd,n=5)
# ---- Taking out BMXBMI
cardioPlusBMIInd2 = select(cardioPlusBMIInd, -2)
# head(cardioPlusBMIInd2, n=5)

# ---- Dichotomize AGE at c=21 for Individual Analysis
# ---- AGE < 21 = 0 and AGE >= 21 = 1
AGEdInd=rep(0,length(RIDAGEYR));AGEdInd[RIDAGEYR>=21]=1;
cardioPlusAGEInd<-data.frame(cardioPlus6,AGEdInd)
# head(cardioPlusAGEInd,n=5)
# ---- Taking out RIDAGEYR
cardioPlusAGEInd2 = select(cardioPlusAGEInd, -3)
# head(cardioPlusAGEInd2, n=5)

# ---- Dichotomize NGS at c=0.38 for Combination Analysis
# ---- NGS > 0.38 = 0 and NGS <= 0.38 = 1
NGSd=rep(0,length(NGS));NGSd[NGS<=0.38]=1;
cardioPlus7<-data.frame(cardioPlus6,NGSd)
# head(cardioPlus7,n=5)
# ---- Taking out NGS
cardioPlus8 = select(cardioPlus7, -10)
# head(cardioPlus8, n=5)

# ---- Dichotomize BMI at c=27 for Combination Analysis
# ---- BMI < 27 = 0 and BMI >= 27 = 1
BMId=rep(0,length(BMXBMI));BMId[BMXBMI>=27]=1;
```

```
cardioPlus9<-data.frame(cardioPlus8,BMId)
# head(cardioPlus9,n=5)
# ---- Taking out BMXBMI
cardioPlus10 = select(cardioPlus9, -2)
# head(cardioPlus10, n=5)

# ---- Dichotomize AGE at c=18 for Combination Analysis
# ---- AGE < 18 = 0 and AGE >= 18 = 1
AGEd=rep(0,length(RIDAGEYR));AGEd[RIDAGEYR>=18]=1;
cardioPlus11<-data.frame(cardioPlus10,AGEd)
# head(cardioPlus11,n=5)
# ---- Taking out RIDAGEYR
cardioPlus12 = select(cardioPlus11, -2)
# head(cardioPlus12, n=5)

# ---- Set generic data frame for Individual Analysis
dfindngs <- data.frame(cardioPlusNGSInd2)
dfindbmi <- data.frame(cardioPlusBMIInd2)
dfindage <- data.frame(cardioPlusAGEInd2)

# ---- Set generic data frame for Combination Analysis
df <- data.frame(cardioPlus12)

# ---- Individual Analysis for NGS at c=0.36
DTestNGS=rep(0,length(NGS));DTestNGS[dfindngs$NGSdInd == '1']=1;
cardioPlusTNGS<-data.frame(cardioPlusNGSInd2,DTestNGS)
# head(cardioPlusTNGS,n=5)
NumNGS=rep(0,length(NGS));NumNGS[(ZBIN==1)&(DTestNGS==1)]=1;
DenNGS=rep(0,length(NGS));DenNGS[(ZBIN==1)]=1;
cardioPlusTBNGS<-data.frame(cardioPlusTNGS,NumNGS,DenNGS)
NHANES_designDSensNGS <- svydesign(data=cardioPlusTBNGS, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SensitNGS=svyratio(~NumNGS,~DenNGS,design=NHANES_designDSensNGS)
NumNGS1=rep(0,length(NGS));NumNGS1[(ZBIN==0)&(DTestNGS==0)]=1;
DenNGS1=rep(0,length(NGS));DenNGS1[(ZBIN==0)]=1;
cardioPlusTBNGS1<-data.frame(cardioPlusTNGS,NumNGS1,DenNGS1)
NHANES_designDSpecNGS <- svydesign(data=cardioPlusTBNGS1, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SpecifNGS=svyratio(~NumNGS1,~DenNGS1,design=NHANES_designDSpecNGS)
NHANES_designDNGS <- svydesign(data=cardioPlusTNGS, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
```

```
DTestNGSmean = svymean(DTestNGS,design=NHANES_designDNGS)
ZBINNGSmean = svymean(ZBIN,design=NHANES_designDNGS)


# ---- Individual Analysis for BMI at c=27
DTestBMI=rep(0,length(BMXBMI));DTestBMI[dfindbmi$BMIdInd == '1']=1;
cardioPlusTBMI<-data.frame(cardioPlusBMIInd2,DTestBMI)
# head(cardioPlusTBMI,n=5)
NumBMI=rep(0,length(BMXBMI));NumBMI[(ZBIN==1)&(DTestBMI==1)]=1;
DenBMI=rep(0,length(BMXBMI));DenBMI[(ZBIN==1)]=1;
cardioPlusTBBMI<-data.frame(cardioPlusTBMI,NumBMI,DenBMI)
NHANES_designDSensBMI <- svydesign(data=cardioPlusTBBMI, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SensitBMI=svyratio(~NumBMI,~DenBMI,design=NHANES_designDSensBMI)
NumBMI1=rep(0,length(BMXBMI));NumBMI1[(ZBIN==0)&(DTestBMI==0)]=1;
DenBMI1=rep(0,length(BMXBMI));DenBMI1[(ZBIN==0)]=1;
cardioPlusTBBMI1<-data.frame(cardioPlusTBMI,NumBMI1,DenBMI1)
NHANES_designDSpecBMI <- svydesign(data=cardioPlusTBBMI1, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SpecifBMI=svyratio(~NumBMI1,~DenBMI1,design=NHANES_designDSpecBMI)
NHANES_designDBMI <- svydesign(data=cardioPlusTBMI, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTestBMImean = svymean(DTestBMI,design=NHANES_designDBMI)
ZBINBMImean = svymean(ZBIN,design=NHANES_designDBMI)

# ---- Individual Analysis for AGE at c=21
DTestAGE=rep(0,length(RIDAGEYR));DTestAGE[dfindage$AGEdInd == '1']=1;
cardioPlusTAGE<-data.frame(cardioPlusAGEInd2,DTestAGE)
# head(cardioPlusTAGE,n=5)
NumAGE=rep(0,length(RIDAGEYR));NumAGE[(ZBIN==1)&(DTestAGE==1)]=1;
DenAGE=rep(0,length(RIDAGEYR));DenAGE[(ZBIN==1)]=1;
cardioPlusTBAGE<-data.frame(cardioPlusTAGE,NumAGE,DenAGE)
NHANES_designDSensAGE <- svydesign(data=cardioPlusTBAGE, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
SensitAGE=svyratio(~NumAGE,~DenAGE,design=NHANES_designDSensAGE)
NumAGE1=rep(0,length(RIDAGEYR));NumAGE1[(ZBIN==0)&(DTestAGE==0)]=1;
DenAGE1=rep(0,length(RIDAGEYR));DenAGE1[(ZBIN==0)]=1;
cardioPlusTBAGE1<-data.frame(cardioPlusTAGE,NumAGE1,DenAGE1)
NHANES_designDSpecAGE <- svydesign(data=cardioPlusTBAGE1, id=~SDMVPSU,
strata=~SDMVSTRA,
```

```
weights=~WTMEC2YR, nest=TRUE)
SpecifAGE=svyratio(~NumAGE1,~DenAGE1,design=NHANES_designDSpecAGE)
NHANES_designDAGE <- svydesign(data=cardioPlusTAGE, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTestAGEmean = svymean(DTestAGE,design=NHANES_designDAGE)
ZBINAGEmean = svymean(ZBIN,design=NHANES_designDAGE)

# ---- Select where NGSd=1 AND BMId=1 AND AGEd=1 (Combination Analysis -
Combination 1)
DTest1=rep(0,length(NGS));DTest1[df$NGSd == '1' & df$BMId == '1' & df$AGEd ==
'1']=1;
cardioPlusT1<-data.frame(cardioPlus12,DTest1)
# head(cardioPlusT1,n=5)
Num1=rep(0,length(NGS));Num1[(ZBIN==1)&(DTest1==1)]=1;
Den1=rep(0,length(NGS));Den1[(ZBIN==1)]=1;
cardioPlusTB1<-data.frame(cardioPlusT1,Num1,Den1)
NHANES_designDSens1 <- svydesign(data=cardioPlusTB1, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Sensit1=svyratio(~Num1,~Den1,design=NHANES_designDSens1)
Num11=rep(0,length(NGS));Num11[(ZBIN==0)&(DTest1==0)]=1;
Den11=rep(0,length(NGS));Den11[(ZBIN==0)]=1;
cardioPlusTB11<-data.frame(cardioPlusT1,Num11,Den11)
NHANES_designDSpec1 <- svydesign(data=cardioPlusTB11, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Specif1=svyratio(~Num11,~Den11,design=NHANES_designDSpec1)
NHANES_designD1 <- svydesign(data=cardioPlusT1, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTest1mean = svymean(DTest1,design=NHANES_designD1)
ZBIN1mean = svymean(ZBIN,design=NHANES_designD1)

# ---- Select where NGSd=1 AND (BMId=1 or AGEd=1) (Combination Analysis -
Combination 2)
DTest2=rep(0,length(NGS));DTest2[df$NGSd == '1' & (df$BMId == '1' | df$AGEd ==
'1')]=1;
cardioPlusT2<-data.frame(cardioPlus12,DTest2)
# head(cardioPlusT2,n=5)
Num2=rep(0,length(NGS));Num2[(ZBIN==1)&(DTest2==1)]=1;
Den2=rep(0,length(NGS));Den2[(ZBIN==1)]=1;
cardioPlusTB2<-data.frame(cardioPlusT2,Num2,Den2)
```

```
NHANES_designDSens2 <- svydesign(data=cardioPlusTB2, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Sensit2=svyratio(~Num2,~Den2,design=NHANES_designDSens2)
Num21=rep(0,length(NGS));Num21[(ZBIN==0)&(DTest2==0)]=1;
Den21=rep(0,length(NGS));Den21[(ZBIN==0)]=1;
cardioPlusTB21<-data.frame(cardioPlusT2,Num21,Den21)
NHANES_designDSpec2 <- svydesign(data=cardioPlusTB21, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Specif2=svyratio(~Num21,~Den21,design=NHANES_designDSpec2)
NHANES_designD2 <- svydesign(data=cardioPlusT2, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTest2mean = svymean(DTest2,design=NHANES_designD2)
ZBIN2mean = svymean(ZBIN,design=NHANES_designD2)

# ---- Select where BMId=1 AND (NGSd=1 or AGEd=1) (Combination Analysis -
Combination 3)
DTest3=rep(0,length(NGS));DTest3[df$BMId == '1' & (df$NGSd == '1' | df$AGEd ==
'1')]=1;
cardioPlusT3<-data.frame(cardioPlus12,DTest3)
# head(cardioPlusT3,n=5)
Num3=rep(0,length(NGS));Num3[(ZBIN==1)&(DTest3==1)]=1;
Den3=rep(0,length(NGS));Den3[(ZBIN==1)]=1;
cardioPlusTB3<-data.frame(cardioPlusT3,Num3,Den3)
NHANES_designDSens3 <- svydesign(data=cardioPlusTB3, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Sensit3=svyratio(~Num3,~Den3,design=NHANES_designDSens3)
Num31=rep(0,length(NGS));Num31[(ZBIN==0)&(DTest3==0)]=1;
Den31=rep(0,length(NGS));Den31[(ZBIN==0)]=1;
cardioPlusTB31<-data.frame(cardioPlusT3,Num31,Den31)
NHANES_designDSpec3 <- svydesign(data=cardioPlusTB31, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Specif3=svyratio(~Num31,~Den31,design=NHANES_designDSpec3)
NHANES_designD3 <- svydesign(data=cardioPlusT3, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTest3mean = svymean(DTest3,design=NHANES_designD3)
ZBIN3mean = svymean(ZBIN,design=NHANES_designD3)
```

```
# ---- Select where AGEd=1 AND (NGSd=1 or BMId=1) (Combination Analysis - Combination 4)
DTest4=rep(0,length(NGS));DTest4[df$AGEd == '1' & (df$NGSd == '1' | df$BMId == '1')]=1;
cardioPlusT4<-data.frame(cardioPlus12,DTest4)
# head(cardioPlusT4,n=5)
Num4=rep(0,length(NGS));Num4[(ZBIN==1)&(DTest4==1)]=1;
Den4=rep(0,length(NGS));Den4[(ZBIN==1)]=1;
cardioPlusTB4<-data.frame(cardioPlusT4,Num4,Den4)
NHANES_designDSens4 <- svydesign(data=cardioPlusTB4, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Sensit4=svyratio(~Num4,~Den4,design=NHANES_designDSens4)
Num41=rep(0,length(NGS));Num41[(ZBIN==0)&(DTest4==0)]=1;
Den41=rep(0,length(NGS));Den41[(ZBIN==0)]=1;
cardioPlusTB41<-data.frame(cardioPlusT4,Num41,Den41)
NHANES_designDSpec4 <- svydesign(data=cardioPlusTB41, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Specif4=svyratio(~Num41,~Den41,design=NHANES_designDSpec4)
NHANES_designD4 <- svydesign(data=cardioPlusT4, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTest4mean = svymean(DTest4,design=NHANES_designD4)
ZBIN4mean = svymean(ZBIN,design=NHANES_designD4)

# ---- Select where AGEd=1 or NGSd=1 or BMId=1 (Combination Analysis - Combination 5)
DTest5=rep(0,length(NGS));DTest5[(df$NGSd == '1' | df$BMId == '1' | df$AGEd == '1')]=1;
cardioPlusT5<-data.frame(cardioPlus12,DTest5)
# head(cardioPlusT5,n=5)
Num5=rep(0,length(NGS));Num5[(ZBIN==1)&(DTest5==1)]=1;
Den5=rep(0,length(NGS));Den5[(ZBIN==1)]=1;
cardioPlusTB5<-data.frame(cardioPlusT5,Num5,Den5)
NHANES_designDSens5 <- svydesign(data=cardioPlusTB5, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Sensit5=svyratio(~Num5,~Den5,design=NHANES_designDSens5)
Num51=rep(0,length(NGS));Num51[(ZBIN==0)&(DTest5==0)]=1;
Den51=rep(0,length(NGS));Den51[(ZBIN==0)]=1;
cardioPlusTB51<-data.frame(cardioPlusT5,Num51,Den51)
NHANES_designDSpec5 <- svydesign(data=cardioPlusTB51, id=~SDMVPSU,
strata=~SDMVSTRA,
```

```
weights=~WTMEC2YR, nest=TRUE)
Specif5=svyratio(~Num51,~Den51,design=NHANES_designDSpec5)
NHANES_designD5 <- svydesign(data=cardioPlusT5, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTest5mean = svymean(DTest5,design=NHANES_designD5)
ZBIN5mean = svymean(ZBIN,design=NHANES_designD5)


# ---- Select where BMId=1 or (NGSd=1 and AGEd=1) (Combination Analysis -
Combination 6)
DTest6=rep(0,length(NGS));DTest6[df$BMId == '1' | (df$NGSd == '1' & df$AGEd ==
'1')]=1;
cardioPlusT6<-data.frame(cardioPlus12,DTest6)
# head(cardioPlusT6,n=5)
Num6=rep(0,length(NGS));Num6[(ZBIN==1)&(DTest6==1)]=1;
Den6=rep(0,length(NGS));Den6[(ZBIN==1)]=1;
cardioPlusTB6<-data.frame(cardioPlusT6,Num6,Den6)
NHANES_designDSens6 <- svydesign(data=cardioPlusTB6, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Sensit6=svyratio(~Num6,~Den6,design=NHANES_designDSens6)
Num61=rep(0,length(NGS));Num61[(ZBIN==0)&(DTest6==0)]=1;
Den61=rep(0,length(NGS));Den61[(ZBIN==0)]=1;
cardioPlusTB61<-data.frame(cardioPlusT6,Num61,Den61)
NHANES_designDSpec6 <- svydesign(data=cardioPlusTB61, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Specif6=svyratio(~Num61,~Den61,design=NHANES_designDSpec6)
NHANES_designD6 <- svydesign(data=cardioPlusT6, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTest6mean = svymean(DTest6,design=NHANES_designD6)
ZBIN6mean = svymean(ZBIN,design=NHANES_designD6)

# ---- Select where NGSd=1 or (BMId=1 and AGEd=1) (Combination Analysis -
Combination 7)
DTest7=rep(0,length(NGS));DTest7[df$NGSd == '1' | (df$BMId == '1' & df$AGEd ==
'1')]=1;
cardioPlusT7<-data.frame(cardioPlus12,DTest7)
# head(cardioPlusT7,n=5)
Num7=rep(0,length(NGS));Num7[(ZBIN==1)&(DTest7==1)]=1;
Den7=rep(0,length(NGS));Den7[(ZBIN==1)]=1;
cardioPlusTB7<-data.frame(cardioPlusT7,Num7,Den7)
```

```
NHANES_designDSens7 <- svydesign(data=cardioPlusTB7, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Sensit7=svyratio(~Num7,~Den7,design=NHANES_designDSens7)
Num71=rep(0,length(NGS));Num71[(ZBIN==0)&(DTest7==0)]=1;
Den71=rep(0,length(NGS));Den71[(ZBIN==0)]=1;
cardioPlusTB71<-data.frame(cardioPlusT7,Num71,Den71)
NHANES_designDSpec7 <- svydesign(data=cardioPlusTB71, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Specif7=svyratio(~Num71,~Den71,design=NHANES_designDSpec7)
NHANES_designD7 <- svydesign(data=cardioPlusT7, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTest7mean = svymean(DTest7,design=NHANES_designD7)
ZBIN7mean = svymean(ZBIN,design=NHANES_designD7)

# ---- Select where AGEd=1 or (NGSd=1 and BMId=1) (Combination Analysis -
Combination 8)
DTest8=rep(0,length(NGS));DTest8[df$AGEd == '1' | (df$NGSd == '1' & df$BMId ==
'1')]=1;
cardioPlusT8<-data.frame(cardioPlus12,DTest8)
# head(cardioPlusT8,n=5)
Num8=rep(0,length(NGS));Num8[(ZBIN==1)&(DTest8==1)]=1;
Den8=rep(0,length(NGS));Den8[(ZBIN==1)]=1;
cardioPlusTB8<-data.frame(cardioPlusT8,Num8,Den8)
NHANES_designDSens8 <- svydesign(data=cardioPlusTB8, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Sensit8=svyratio(~Num8,~Den8,design=NHANES_designDSens8)
Num81=rep(0,length(NGS));Num81[(ZBIN==0)&(DTest8==0)]=1;
Den81=rep(0,length(NGS));Den81[(ZBIN==0)]=1;
cardioPlusTB81<-data.frame(cardioPlusT8,Num81,Den81)
NHANES_designDSpec8 <- svydesign(data=cardioPlusTB81, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
Specif8=svyratio(~Num81,~Den81,design=NHANES_designDSpec8)
NHANES_designD8 <- svydesign(data=cardioPlusT8, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
DTest8mean = svymean(DTest8,design=NHANES_designD8)
ZBIN8mean = svymean(ZBIN,design=NHANES_designD8)

# ---- CI's for Individual Analysis
```

```
df = 61-29
CINGSlower = mean(DTestNGSmean) - qt(0.975,df)*SE(DTestNGSmean)
CINGSupper = mean(DTestNGSmean) + qt(0.975,df)*SE(DTestNGSmean)
CIBMIlower = mean(DTestBMImean) - qt(0.975,df)*SE(DTestBMImean)
CIBMIupper = mean(DTestBMImean) + qt(0.975,df)*SE(DTestBMImean)
CIAGElower = mean(DTestAGEmean) - qt(0.975,df)*SE(DTestAGEmean)
CIAGEupper = mean(DTestAGEmean) + qt(0.975,df)*SE(DTestAGEmean)

CINGSlower;CINGSupper
CIBMIlower;CIBMIupper
CIAGElower;CIAGEupper
# ---- CI's for DTest (Combination Analysis)
df = 61-29
CIDTest1lower = mean(DTest1mean) - qt(0.975,df)*SE(DTest1mean)
CIDTest1upper = mean(DTest1mean) + qt(0.975,df)*SE(DTest1mean)
CIDTest2lower = mean(DTest2mean) - qt(0.975,df)*SE(DTest2mean)
CIDTest2upper = mean(DTest2mean) + qt(0.975,df)*SE(DTest2mean)
CIDTest3lower = mean(DTest3mean) - qt(0.975,df)*SE(DTest3mean)
CIDTest3upper = mean(DTest3mean) + qt(0.975,df)*SE(DTest3mean)
CIDTest4lower = mean(DTest4mean) - qt(0.975,df)*SE(DTest4mean)
CIDTest4upper = mean(DTest4mean) + qt(0.975,df)*SE(DTest4mean)
CIDTest5lower = mean(DTest5mean) - qt(0.975,df)*SE(DTest5mean)
CIDTest5upper = mean(DTest5mean) + qt(0.975,df)*SE(DTest5mean)
CIDTest6lower = mean(DTest6mean) - qt(0.975,df)*SE(DTest6mean)
CIDTest6upper = mean(DTest6mean) + qt(0.975,df)*SE(DTest6mean)
CIDTest7lower = mean(DTest7mean) - qt(0.975,df)*SE(DTest7mean)
CIDTest7upper = mean(DTest7mean) + qt(0.975,df)*SE(DTest7mean)
CIDTest8lower = mean(DTest8mean) - qt(0.975,df)*SE(DTest8mean)
CIDTest8upper = mean(DTest8mean) + qt(0.975,df)*SE(DTest8mean)

CIDTest1lower;CIDTest1upper
CIDTest2lower;CIDTest2upper
CIDTest3lower;CIDTest3upper
CIDTest4lower;CIDTest4upper
CIDTest5lower;CIDTest5upper
CIDTest6lower;CIDTest6upper
CIDTest7lower;CIDTest7upper
CIDTest8lower;CIDTest8upper

# ---- CI's for ZBIN (Combination Analysis)
CIZBINlower = mean(ZBIN1mean) - qt(0.975,df)*SE(ZBIN1mean)
CIZBINupper = mean(ZBIN1mean) + qt(0.975,df)*SE(ZBIN1mean)
CIZBINlower;CIZBINupper
```

```
# ---- Print Sensitivity ratios and SD's (Individual Analysis)
SensitNGS
SensitBMI
SensitAGE

# ---- Print Sensitivity ratios and SD's (Combination Analysis)
Sensit1
Sensit2
Sensit3
Sensit4
Sensit5
Sensit6
Sensit7
Sensit8

# ---- Print Specificity ratios and SD's (Individual Analysis)
SpecifNGS
SpecifBMI
SpecifAGE

# ---- Print Specificity ratios and SD's (Combination Analysis)
Specif1
Specif2
Specif3
Specif4
Specif5
Specif6
Specif7
Specif8
```

SAS Code for Individual and Combination Confidence Intervals and Sensitivity/Specificity Levels (Sections 4.3, 4.4, and 5.4)

```
## Code continued from Section 2.3 SAS code ##

/* Dichotomize NGS at c=0.36 - Individual Analysis */
data combinedhealthy;
set combinedhealthy;
NGSdInd=NGS;
if NGS <= 0.36 then NGSdInd = 1;
else NGSdInd = 0;
run;

/* Dichotomize BMI at c=27 - Individual Analysis */
```

```
data combinedhealthy;
set combinedhealthy;
BMIdInd=BMXBMI;
if BMXBMI >= 27 then BMIdInd = 1;
else BMIdInd = 0;
run;

/* Dichotomize AGE at c=21 - Individual Analysis */
data combinedhealthy;
set combinedhealthy;
AGEdInd=RIDAGEYR;
if RIDAGEYR >= 21 then AGEdInd = 1;
else AGEdInd = 0;
run;

/* Dichotomize NGS at c=0.38 - Combination Analysis */
data combinedhealthy;
set combinedhealthy;
NGSd=NGS;
if NGS <= 0.38 then NGSd = 1;
else NGSd = 0;
run;

/* Dichotomize BMI at c=27 - Combination Analysis */
data combinedhealthy;
set combinedhealthy;
BMId=BMXBMI;
if BMXBMI >= 27 then BMId = 1;
else BMId = 0;
run;

/* Dichotomize AGE at c=18 - Combination Analysis */
data combinedhealthy;
set combinedhealthy;
AGEd=RIDAGEYR;
if RIDAGEYR >= 18 then AGEd = 1;
else AGEd = 0;
run;

/* Specify DTestNGS, etc... - Individual Analysis*/
data combinedhealthy;
set combinedhealthy;
if NGSdInd=1 then DTestNGS=1;
else DTestNGS=0;
```

```
run;

data combinedhealthy;
set combinedhealthy;
if BMIdInd=1 then DTestBMI=1;
else DTestBMI=0;
run;

data combinedhealthy;
set combinedhealthy;
if AGEdInd=1 then DTestAGE=1;
else DTestAGE=0;
run;


/* Specify DTest1, etc... - Combination Analysis*/
data combinedhealthy;
set combinedhealthy;
if NGSd=1 and BMId=1 and AGEd=1 then DTEST1=1;
else DTest1=0;
run;

data combinedhealthy;
set combinedhealthy;
if NGSd=1 and (BMId=1 or AGEd=1) then DTEST2=1;
else DTest2=0;
run;

data combinedhealthy;
set combinedhealthy;
if BMId=1 and (NGSd=1 or AGEd=1) then DTEST3=1;
else DTest3=0;
run;

data combinedhealthy;
set combinedhealthy;
if AGEd=1 and (NGSd=1 or BMId=1) then DTEST4=1;
else DTest4=0;
run;

data combinedhealthy;
set combinedhealthy;
if BMId=1 or NGSd=1 or AGEd=1 then DTEST5=1;
else DTest5=0;
```

```
run;

data combinedhealthy;
set combinedhealthy;
if BMId=1 or (NGSd=1 and AGEd=1) then DTEST6=1;
else DTest6=0;
run;

data combinedhealthy;
set combinedhealthy;
if NGSd=1 or (BMId=1 and AGEd=1) then DTEST7=1;
else DTest7=0;
run;

data combinedhealthy;
set combinedhealthy;
if AGEd=1 or (NGSd=1 and BMId=1) then DTEST8=1;
else DTest8=0;
run;

proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTestNGS';
        class DTestNGS ZBIN;
        var DTestNGS ZBIN;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTestBMI';
        class DTestBMI ZBIN;
        var DTestBMI ZBIN;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTestAGE';
        class DTestAGE ZBIN;
        var DTestAGE ZBIN;
```

```
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;


proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTest1';
        class DTest1 ZBIN;
        var DTest1 ZBIN;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTest2';
        class DTest2 ZBIN;
        var DTest2 ZBIN;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTest3';
        class DTest3 ZBIN;
        var DTest3 ZBIN;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTest4';
        class DTest4 ZBIN;
        var DTest4 ZBIN;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
```

```
        ODS GRAPHICS OFF;
run;

proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTest5';
        class DTest5 ZBIN;
        var DTest5 ZBIN;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTest6';
        class DTest6 ZBIN;
        var DTest6 ZBIN;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTest7';
        class DTest7 ZBIN;
        var DTest7 ZBIN;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveymeans data=combinedhealthy;
title 'Survey Means/CI DTest8';
        class DTest8 ZBIN;
        var DTest8 ZBIN;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveyfreq data=combinedhealthy;
```

```
title 'Sensitivity and Specificity Individual NGS';
        tables DTestNGS*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveyfreq data=combinedhealthy;
title 'Sensitivity and Specificity Individual BMI';
        tables DTestBMI*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveyfreq data=combinedhealthy;
title 'Sensitivity and Specificity Individual AGE';
        tables DTestAGE*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveyfreq data=combinedhealthy;
title 'Sensitivity and Specificity Combination 1';
        tables DTest1*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveyfreq data=combinedhealthy;
title 'Sensitivity and Specificity Combination 2';
        tables DTest2*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;
```

```
proc surveyfreq data=combinedhealthy;
title 'Sensitivity and Specificity Combination 3';
        tables DTest3*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveyfreq data=combinedhealthy;
title 'Sensitivity and Specificity Combination 4';
        tables DTest4*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveyfreq data=combinedhealthy;
title 'Sensitivity and Specificity Combination 5';
        tables DTest5*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveyfreq data=combinedhealthy;
title 'Sensitivity and Specificity Combination 6';
        tables DTest6*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;

proc surveyfreq data=combinedhealthy;
title 'Sensitivity and Specificity Combination 7';
        tables DTest7*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;
```

```
proc surveyfreq data=combinedhealthy;
title 'Sensitivity and Specificity Combination 8';
        tables DTest8*ZBIN / SENSPEC;
        strata  SDMVSTRA;
        cluster SDMVPSU;
        weight  WTMEC2YR;
        ODS GRAPHICS OFF;
run;
```

Chapter 5

Section 5.3: SRS Simulation Study (repeated/adjusted for Clustering and Stratification)

```
# system.time(source("Simulations_SRS.r"))

rm(list=ls())

library(lhs)
library(DiceKriging)
library(DiceOptim)
library(survey)
library(knitr)
library(rgl)
library(MASS)

################     LOGISTIC SIMULATIONS  ###################

# these are the true regression coefficients (you can change these, if you want)
bet0=-5;bet=c(4, -2, 3, 0.1, -0.01);

# N is sample size, Nrep is the number of simulations
N=1000;Nrep=100;

# Equal weights for SRS
Wghts<-100*rep(1,N);

# Creates a N x 5 regression matrix, where 5 is the number of 'bet' coeffs above
Rsig=1;X=matrix(Rsig*rnorm(N*length(bet)),N,length(bet));

# Computes probabilities of success for each of the N subjects, using logit link
Probsf=exp(bet0+X%*%bet)/(1+exp(bet0+X%*%bet));
```

```
# this is where the estimated regression coefficients will be collected
SimCoef=matrix(NA,Nrep,1+length(bet))

# repeats the simulation process Nrep times
for (rep in 1:Nrep){
# simulates binary data, as response variable, from the probabilities above
Yf=rep(NA,N);for (i in 1:N){Yf[i]=rbinom(1, 1, Probsf[i])}
Dat_str=data.frame(X,Yf,Wghts)
# Fits the logistic regression model
Lro_design <- svydesign(data=Dat_str, id=~1, weights=~Wghts)
Lro=svyglm(I(Yf==1)~X1+X2+X3+X4+X5,design = Lro_design,family=binomial(link
= "logit"))
# these are the estimated regression coeffs from the fitted regression model
SimCoef[rep,]=Lro$coef
}

# Does the method work? Check if estimated regression coeffs match the true ones above
# specifically, look at their means and medians over the Nrep simulations
print(c(bet0,bet))
print(apply(SimCoef,2,mean))
print(apply(SimCoef,2,sd))
print(apply(SimCoef,2,median))
print(apply(SimCoef,2,mad))
dev.new();par(mfrow=c(2,3));
hist(SimCoef[,1]);hist(SimCoef[,2]);hist(SimCoef[,3]);
hist(SimCoef[,4]);hist(SimCoef[,5]);hist(SimCoef[,6]);

# create grid bounds
Lo1=round(quantile(X[,1],.01),2);Up1=round(quantile(X[,1],.99),2);
Lo2=round(quantile(X[,2],.01),2);Up2=round(quantile(X[,2],.99),2);
Lo3=round(quantile(X[,3],.01),2);Up3=round(quantile(X[,3],.99),2);
N1=length(seq(Lo1,Up1,.01));N2=length(seq(Lo2,Up2,.01));N3=length(seq(Lo3,Up3,.01
));

# create cut-points for the first 3 columns of X eg at quartiles
Xc=X;
X1=rep(0,N);C1=round(quantile(X[,1],.25),2);X1[X[,1]>=C1]=1;Xc[,1]=X1;
X2=rep(0,N);C2=round(quantile(X[,2],.50),2);X2[X[,2]<=C2]=1;Xc[,2]=X2;
X3=rep(0,N);C3=round(quantile(X[,3],.75),2);X3[X[,3]>=C3]=1;Xc[,3]=X3;

# Calculates logit function for probability N times
Probs=exp(bet0+Xc%*%bet)/(1+exp(bet0+Xc%*%bet));
```

```
# Sets up empty matrix of Nrep rows with 5 columns
RezSim=matrix(NA,Nrep,5)

# Sets up a loop that generates 'Npts' inputs
# each in the input space
for (rep in 1:Nrep){
Y=rep(NA,N);for (i in 1:N){Y[i]=rbinom(1, 1, Probs[i])}
Npts=30;
d <- 3
design<-matrix(1,Npts, d)
while(min(dist(design))==0){
design<-optimumLHS(Npts, d)
# Scales the inputs to the actual bounds
design[,1]=Lo1+ (Up1-Lo1)*design[,1];design[,1]=round(design[,1],2);
design[,2]=Lo2+ (Up2-Lo2)*design[,2];design[,2]=round(design[,2],2);
design[,3]=Lo3+ (Up3-Lo3)*design[,3];design[,3]=round(design[,3],2);
}
n<-dim(design)[1]
design<-data.frame(design)
names(design) <- c("x1", "x2", "x3")
apply(design,2,max);apply(design,2,min)

# Computes the AIC for the logistic regression model
# where the cutpoints are dichotomized sequentially
# at the input values
response.AIC=rep(NA,n);
for (i in 1:n){
Xd1=rep(0,N);Xd1[X[,1]>=design[i,1]]=1;
Xd2=rep(0,N);Xd2[X[,2]<=design[i,2]]=1;
Xd3=rep(0,N);Xd3[X[,3]>=design[i,3]]=1;
Xd4=X[,4];Xd5=X[,5];

Dat_strD=data.frame(Xd1,Xd2,Xd3,Xd4,Xd5,Y,Wghts);
Lro_design <- svydesign(data=Dat_strD, id=~1, weights=~Wghts)
LroD=svyglm(I(Y==1)~Xd1+Xd2+Xd3+Xd4+Xd5,design =
Lro_design,family=binomial(link = "logit"))
Lro_stepD<-stepAIC(LroD, trace = FALSE, scope = list(lower = ~ Xd1+Xd2+Xd3,
upper = LroD))
response.AIC[i]=AIC(Lro_stepD)[2]
}

# Defines the AIC function for the logistic regression model
# where the three cutpoints are dichotomized at an arbitrary 'inp'
# transformed on the original scale
```

```
AICf<-function(inp){
Xf1=rep(0,N);Xf1[X[,1]>=inp[1]]=1;
Xf2=rep(0,N);Xf2[X[,2]<=inp[2]]=1;
Xf3=rep(0,N);Xf3[X[,3]>=inp[3]]=1;Xf4=X[,4];Xf5=X[,5];
Dat_str_f=data.frame(Xf1,Xf2,Xf3,Xf4,Xf5,Y,Wghts);
Lro_design <- svydesign(data=Dat_str_f, id=~1, weights=~Wghts)
Lrof=svyglm(I(Y==1)~Xf1+Xf2+Xf3+Xf4+Xf5,design =
Lro_design,family=binomial(link = "logit"))
Lro_stepf<-stepAIC(Lrof, trace = FALSE, scope = list(lower = ~ Xf1+Xf2+Xf3, upper =
Lrof))
obj.fctn=AIC(Lro_stepf)[2]
}

# Sets up values to be used later
lower <- c(Lo1,Lo2,Lo3);upper <- c(Up1,Up2,Up3);
design.up<-design;Dat.up<-response.AIC;CBFV=min(Dat.up);Val=CBFV+1;
ct<-0;EVstep=CBFV;E1step=0;E2step=0;E3step=0;
E1stepNew=0;E2stepNew=0;E3stepNew=0;RErr=0;
prct<-0.0001;

# Runs a while loop until further improvement in AIC becomes
# too small to be worth continuing the loop and the relative error
# is less than a specified small value
# At each iteration, the input set is augmented with the point
# where the EI is maximized
while(((Val>CBFV)|(abs(Val-CBFV)>prct*abs(CBFV)))&(min(dist(design.up))>0)){
KM.model.EGO<-km(design = design.up, response =
Dat.up,control=list(trace=FALSE),covtype="exp")
CBFV<-min(Dat.up);
# This is the point where EI is maximized at that iteration of the loop
MEI<-max_EI(KM.model.EGO, lower=lower, upper=upper);
MEI.par=MEI$par;MEI.par[,1]=round(MEI.par[,1],2);
MEI.par[,2]=round(MEI.par[,2],2);MEI.par[,3]=round(MEI.par[,3],2);
# augments the previous input set with the above point
design.up<-t(matrix(c(c(t(as.matrix(design.up))),c(MEI.par)),d,dim(design.up)[1]+1))
design.up<- data.frame(design.up);names(design.up) <- c("x1", "x2", "x3")
Val=AICf(MEI.par);
# Transforms the new point on the original scale
E1stepNew=MEI.par[1];
E2stepNew=MEI.par[2];
E3stepNew=MEI.par[3];
# Updates the values, and closes the loop
Dat.up<-c(Dat.up,Val);RErr<-c(RErr,abs(Val-CBFV)/abs(CBFV));
EVstep=c(EVstep,Val);E1step=c(E1step,E1stepNew);
```

```
E2step=c(E2step,E2stepNew);E3step=c(E3step,E3stepNew);
}

# Fills the simulation matrix with the updated values, number of iterations
# of the EGO algorithm, and the AIC at last iteration
RezSim[rep,]=c(E1stepNew,E2stepNew,E3stepNaew,length(E1step),Val)
print(rep)
}

# Calculates mean SD
SDm<-function(x){mean(abs(x-mean(x)))}
print(c(bet0,bet))
print(apply(SimCoef,2,mean))
print(c(apply(SimCoef,2,SDm)))
print(apply(SimCoef,2,sd))
print(apply(SimCoef,2,median))
print(apply(SimCoef,2,mad))
dev.new();par(mfrow=c(2,3));
hist(SimCoef[,1]);hist(SimCoef[,2]);hist(SimCoef[,3]);
hist(SimCoef[,4]);hist(SimCoef[,5]);hist(SimCoef[,6]);

#True cut-points
print(c(C1,C2,C3))
# Mean values of the simulations
print(c(apply(RezSim,2,mean)))
# Average of the standard deviations for all simulations
print(c(apply(RezSim,2,SDm)))
# Standard deviation for all simulations
print(apply(RezSim,2,sd))
# Median values of the simulations
print(apply(RezSim,2,median))
# Median absolue deviation
print(apply(RezSim,2,mad))
dev.new();par(mfrow=c(2,3));
hist(RezSim[,1]);hist(RezSim[,2]);hist(RezSim[,3]);
hist(RezSim[,4]);hist(RezSim[,5]);
```

Section 5.4: Figure 5.4.1

```
# system.time(source("NHANES_Kriging_Validation.r"))

rm(list=ls())
```

```
library(lhs)
library(DiceKriging)
library(DiceOptim)
library(survey)
library(rgl)
library(MASS)
library(dplyr)

set.seed(210)

################     LOGISTIC REG DATA       ###################

cardio=read.table("RData9.txt",header=T)
attach(cardio)

race1=rep(0,dim(cardio)[1]);race1[RIDRETH3==1]=1;race1[RIDRETH3==7]=-1;
race2=rep(0,dim(cardio)[1]);race2[RIDRETH3==2]=1;race2[RIDRETH3==7]=-1;
race3=rep(0,dim(cardio)[1]);race3[RIDRETH3==3]=1;race3[RIDRETH3==7]=-1;
race4=rep(0,dim(cardio)[1]);race4[RIDRETH3==4]=1;race4[RIDRETH3==7]=-1;
race6=rep(0,dim(cardio)[1]);race6[RIDRETH3==6]=1;race6[RIDRETH3==7]=-1;
Ind_SV=Ind_svy;Ind_SV[Ind_svy==0]=1;Ind_SV[Ind_svy==1]=-1;
Gend=RIAGENDR;Gend[RIAGENDR==2]=-1;

cardioPlus<-data.frame(cardio,race1,race2,race3,race4,race6,Ind_SV,Gend)

# --- Taking out RIAGENDR, RIDRETH3, Ind_svy, and ZSCORE
cardioPlus2 = select(cardioPlus, -3, -5, -10, -14)

NHANES_design0 <- svydesign(data=cardioPlus2, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)

NHANES_obj<-svyglm(I(ZBIN==1)~NGS+Ind_SV+Gend+RIDAGEYR+BMXBMI
+race1+race2+race3+race4+race6
+PAD680+INDFMPIR+DBD895, design = NHANES_design0, family=binomial(link =
"logit"))

round(coef(summary(NHANES_obj)),4)

################     INITIAL DESIGN     ###################

LoCGS=0.30;UpCGS=0.45;NG=length(seq(LoCGS,UpCGS,.01))
LoBMI=17;UpBMI=37;NB=length(LoBMI:UpBMI)
LoNHM=12;UpNHM=24;NH=length(LoNHM:UpNHM)
```

```
Npts=30;
d <- 3
design<-matrix(1,Npts, d)
while(min(dist(design))==0){
design<-optimumLHS(Npts, d)
design[,1]=LoCGS + (UpCGS-LoCGS)*design[,1];design[,1]=round(design[,1],2);
design[,2]=LoBMI + (UpBMI-LoBMI)*design[,2];design[,2]=round(design[,2]);
design[,3]=LoNHM + (UpNHM-LoNHM)*design[,3];design[,3]=round(design[,3]);
}
n<-dim(design)[1]
design<-data.frame(design)
names(design) <- c("x1", "x2", "x3")
apply(design,2,max)
apply(design,2,min)

Dat=rep(NA,n);

for (i in 1:n){
NGSd=rep(0,length(NGS));NGSd[NGS<=design[i,1]]=1;
BMId=rep(0,length(NGS));BMId[BMXBMI>=design[i,2]]=1;
AGEd=rep(0,length(NGS));AGEd[RIDAGEYR>=design[i,3]]=1;


cardioPlusD<-data.frame(cardioPlus2,NGSd,BMId,AGEd)

NHANES_designD <- svydesign(data=cardioPlusD, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)

NHANES_objD<-svyglm(I(ZBIN==1)~NGSd+Ind_SV+Gend+AGEd+BMId
+race1+race2+race3+race4+race6
+PAD680+INDFMPIR+DBD895, design = NHANES_designD, family=binomial(link =
"logit"))
NHANES_step<-stepAIC(NHANES_objD, trace = FALSE, scope = list(lower = ~
NGSd+BMId+AGEd, upper = NHANES_objD))

Dat[i]=AIC(NHANES_step)[2]
print(i)
}

###############     KRIGING AND OPTIMIZATION    ###################

AICf<-function(ord01){
```

```
NGSd=rep(0,length(NGS));NGSd[NGS <= ord01[1]]=1;
BMId=rep(0,length(NGS));BMId[BMXBMI >= ord01[2]]=1;
AGEd=rep(0,length(NGS));AGEd[RIDAGEYR >= ord01[3]]=1;

cardioPlusD<-data.frame(cardioPlus,NGSd,BMId,AGEd)
NHANES_designD <- svydesign(data=cardioPlusD, id=~SDMVPSU,
strata=~SDMVSTRA,
weights=~WTMEC2YR, nest=TRUE)
NHANES_objD<-svyglm(I(ZSCOREnd==1)~NGSd+Ind_SV+Gend+AGEd+BMId
+race1+race2+race3+race4+race6
+PAD680+INDFMPIR+DBD895, design = NHANES_designD, family=binomial(link =
"logit"))
NHANES_step<-stepAIC(NHANES_objD, trace = FALSE, scope = list(lower = ~
NGSd+BMId+AGEd, upper = NHANES_objD))
obj.fctn=AIC(NHANES_step)[2]
}

KM.model<-km(design = design, response =
Dat,control=list(trace=FALSE),covtype="exp")

plot(KM.model)
```

REFERENCES

Akobeng, A. K. (2007). Understanding diagnostic tests 3: Receiver operating characteristic curves. *Acta Paediatrica*, *96*(5), 644–647. https://doi:10.1111/j.1651-2227.2006.00178.x

American Diabetes Association. (2012). Executive summary: standards of medical care in diabetes−2012. *Diabetes Care*. *35*(Supplement_1). https://doi.org/10.2337/dc12-s004

American Diabetes Association. (2018). 2. classification and diagnosis of diabetes: Standards of medical care in diabetes – 2019. *Diabetes Care*. *42*(Supplemenent_1). https://doi.org/10.2337/dc19-s002

Binder D.A. (1983). On the variances of asymptotically normal estimators from complex surveys. *International Statistical Review*, *51*(3), 279–292. https://doi.org/10.2307/1402588

Brouwer, S. I., Stolk, R. P., Liem, E. T., Lemmink, K. A., & Corpelejin, E. (2012). The role of fitness in the association between fatness and cardiometabolic risk from childhood to adolescence. *Pediatric Diabetes*, *14*(1), 57-65. https://doi.org/10.1111/j.1399-5448.2012.00893.x.

Brown, E. C., Buchan, D. S., Madi, S. A., Gordon, B. N., & Drignei, D. (2020). Grip strength cut points for diabetes risk among apparently healthy U.S. Adults. *American Journal of Preventative Medicine,* 58(6), 757-765. https://doi.org/10.1016/j.amepre.2020.01.016

Cressie N. A. C. (1993). *Statistics for Spatial Data*. John Wiley & Sons, Inc.

DeHondt, B. G., Madi, S. A., Drignei, D., Buchan, D. S., & Brown, E. C. (2022). Handgrip strength cut-points for cardiometabolic risk identification in U.S. younger population. *Measurement in Physical Education and Exercise Science*, 1-10. https://doi.org/10.1080/1091367x.2022.2160254

DeMers, D., & Wachs, D. (n.d.). *Physiology. mean arterial pressure – StatPearls – NCBI Bookshelf.* Retrieved 2022, from https://www.ncbi.nlm.nih.gov/books/NBK538226/

Ferri, C., Hernández-Orallo, J., & Salido, M. A. (2003). Volume under the ROC surface for multi-class problems. *Machine Learning: ECML 2003,* 108-120. https://doi.org/10.1007/978-3-540-39857-8_12

Fleiss, J. L., Levin, B. A., & Paik, M. C. (2003). *Statistical Methods for rates and proportions*. J. Wiley.

Garcia-Hermoso, A., Tordecilla-Sanders, A., Correa-Bautista, J. E., Peterson, M. D., Izquierdo, M., Quino-Ávila, A. C., Sandoval-Cuellar, C., González-Ruiz, K., & Ramírez-Vélez, R. (2020). Muscle strength cut-offs for the detection of metabolic syndrome in a nonrepresentative sample of collegiate students from Colombia. *Journal of Sport and Health Science, 9*(3), 283-290. https://doi.org/10.1016/j.jshs.2018.09.004

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning with applications in R.* Springer.

Janes, H., Longton, G., & Pepe, M. S. (2009). Accommodating covariates in receiver operating characteristic analysis. *The State Journal: Promoting Communications on Statistics and Stata*, *9*(1), 17-39. https://doi.org/10.1177/1536867x0900900102

Johnson C. L., Dohrmann, S. M., Burt, V. L., & Mohadjer, L. K. (2014). *National Health and Nutrition Examination Survey: Sample design, 2011–2014*. Retrieved 2022, from https://pubmed.nbci.nlm.nih/gov/25569458/

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization, 13*(4), 455-492. https://doi.org/10.1023/a:1008306431147

Koyama, T., Hamada, H., Nishida, M., Naess, P. A., Gaarder, C., & Sakamoto, T. (2016). Defining the optimal cut-off values for liver enzymes in diagnosing blunt liver injury. *BMC Research Notes*, *9*(1). https://doi:10.1186/s13104-016-1863-3

Kutner, M. H., Nachtsheim, C., Neter, J., & Li, W. (2005). *Applied Linear Statistical Models* (5th ed.). McGraw-Hill Irwin.

Lee, W. C., Kuo, L. C., Cheng, Y. C., Chen, C.W., Lin, Y.K., Lin, T.Y., & Lin, H. L. (2010). Combination of white blood cell count with liver enzymes in the diagnosis of blunt liver laceration. *The American Journal of Emergency Medicine*, *28*(9), 1024-1029. https://doi.org/10.1016/j.ajem.2009.06.005

Lohr, S. L. (2010). *Sampling design and analysis*. CRC Press, Taylor & Francis Group.

Lumley, T. (n.d.). *Package 'survey'.* Survey: Analysis of Complex Survey Samples. Retrieved 2022, from https://cran.r-project.org/web/packages/survey/survey.pdf

Magge, S. N., Goodman, E., Armstrong, S. C., Daniels, S., Corkins, M., de Ferranti, S., Golden, N. H., Kim, J. H., Schwarzenberg, S. J., Sills, I. N., Casella, S. J., DeMeglio, L. A., Gonzales, J. L., Kaplowitz, P.B., Lynch, J. L., Wintergerst, K. A., Bolling, C.F., …, Schwarz, R. P. (2017). The metabolic syndrome in children and adolescents: Shifting the focus to cardiometabolic risk factor clustering. *Pediatrics*, *140*(2). https://doi.org/10.1542/peds.2017-1603

Morris, M. D., & Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference, 43*(3), 381-402. https://doi.org/10.1016/0378-3758(94)00035-t

Peterson, M. D., Zhang, P., Saltarelli, W. A., Visich, P. S., & Gordon, P. M. (2016). Low muscle strength thresholds for the detection of cardiometabolic risk in adolescents. *American Journal of Preventative Medicine, 50*(5), 593-599. https://doi.org/10.1016/j.amepre.2015.09.019

Ramírez-Vélez, R., Peña-Ibagon, J. C., Martínez-Torres, J., Tordecilla-Sanders, A., Correa-Bautista, J. E., Lobela, F., & García-Hermoso, A. (2017). Handgrip strength cutoff for cardiometabolic risk index among Colombian children and adolescents: The FUPRECOL study. *Scientific Reports, 7*(1). https://doi.org/10.1038/srep42622

Roustant, O., Ginsbourger, D., & Deville, Y. (2012). DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, *51*(1). https://doi.org/10.18637/jss.vs051.io1

Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science, 4*(4). https://doi.org/10.1214/ss/1177012413

Santner, T. J., Williams, B. J., & Notz, W. (2003). *The design and analysis of computer experiments*. Springer.

SAS Institute Inc. (2017). *SAS/STAT® 14.3 User's Guide. Chapter 114*. The SURVEYLOGISTIC Procedure. Cary, NC. Retrieved from https://support.sas.com/documentation/onlinedoc/stat/143/plm.pdf

SAS Institute Inc. (2020). *SAS/STAT® 15.2 User's Guide. Chapter 117*. The SURVEYFREQ Procedure. Cary, NC.

Sukhatme, S., & Beam, C. A. (1994). Stratification in nonparametric roc studies. *Biometrics*, *50*(1), 149. https://doi.org/10.2307/2533205

Tan, K. K., Bang, S.L., Vijayan, A., & Chiu, M. T. (2009). Hepatic enzymes have a role in the diagnosis of hepatic injury after blunt abdominal trauma. *Injury, 40*(9), 978-983. https://doi.org/10.1016/j.injury.2009.02.023

Tian, Z., Liu, H., Su, X., Fang, Z., Dong, Z., Yu, C., & Luo, K. (2012). Role of elevated liver transaminase levels in the diagnosis of liver injury after blunt abdominal trauma. *Experimental and Therapeutic Medicine*, *4*(2), 255-60. https://doi.org/10.3892/etm.2012.575

Yao, W., Li, Z., & Graubard, B. I. (2014). Estimation of ROC curve with complex survey data. *Statistics in Medicine*, *34*(8), 1293–1303. https://doi.org/10.1002/sim.6405

Zou, K. H., O'Malley, A. J., & Mauri, L. (2007). Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models. *Circulation*, *115*(5), 654–657. https://doi.org/10.1161/circulationaha.105.594929